UNIVERSIDAD SAN FRANCISCO DE QUITO

Diseño e Implementación de un sistema electrónico de Alarma e Inmovilización Vehicular controlado por un Teléfono Inteligente con comunicación Bluetooth

Andrés Danilo Galeas Arthos

Luis Miguel Prócel, M.S., director de tesis

Tesis de grado presentada como requisito para la obtención del título de Ingeniero Eléctrico Electrónico en Telecomunicaciones

Quito, febrero del 2013

Universidad San Francisco de Quito
Colegio Politécnico

HOJA DE APROBACION DE TESIS

Diseño e Implementación de un sistema electrónico de Alarma e Inmovilización Vehicular controlado por un Teléfono Inteligente con comunicación Bluetooth

Andrés Danilo Galeas Arthos

Luis Miguel Prócel, M.S. ....………………..………………………
Director de Tesis y
Miembro del Comité de Tesis

Nelson Herrera, Ing. ………………………………….………………
Profesor del Colegio Politécnico y
Miembro del Comité de Tesis

René Játiva, D.E.A. ……………....……………………………………
Profesor del Colegio Politécnico y
Miembro del Comité de Tesis

Omar Aguirre, M.S. ……………………………………………………
Profesor del Colegio Politécnico y
Miembro del Comité de Tesis

Luis Caiza, M.S. ……………………………………………………
Profesor del Colegio Politécnico y
Miembro del Comité de Tesis

Ximena Córdova, Ph.D. ……………………………………………………
Decana del Colegio Politécnico y
Miembro del Comité de Tesis

Quito, febrero del 2013

Andrés Danilo Galeas Arthos
171162815-4

Martes, 26 de febrero del 2013

Dedicatoria

Este proyecto no habría existido sin el conocimiento adquirido durante la carrera universitaria de ingeniería eléctrica electrónica. Las clases que permitieron conjeturar la propuesta inicial y el desarrollo de esta tesis fueron: todas las clases de Programación, los proyectos de Microcontroladores, los capítulos de Control Automático, las enseñas impartidas en Comunicaciones Móviles, y las horas invertidas en los laboratorios de la carrera.

Agradecimientos

Mis primeros agradecimientos son para mis padres Wilson y Mercedes, quienes han sido, son y serán mi ejemplo a seguir en la vida.

A continuación mis agradecimientos a las siguientes personas:

A mi director de tesis y amigo Luis Miguel Procel por brindarme su apoyo para finalizar y presentar esta tesis.

A mi profesor y amigo Nelson Herrera quien me guió a desarrollar este gran logro.

A René Játiva, Omar Aguirre, Luis Caiza y Ximena Córdova por aceptar de manera muy cordial su participación en el Comité de Tesis.

A mi hermana María Belén y mi hermano Santiago a quienes animo a realizar un último esfuerzo y sacrificio personal.

A María José quien me ha brindado su compañía y cariño durante el desarrollo y conclusión de esta tesis.

A Juan Yepes, Aníbal Soliz, Jaime Carbo y María del Carmen Túqueres, quienes me brindaron una mano desinteresada durante la realización e implementación del proyecto.

A mi primo Roberto Jiménez por depositar su confianza en mí y permitirme instalar el dispositivo final en su vehículo.

A mis amigos compañeros de carrera.

Y un agradecimiento especial a Dios Padre, por no dejarme abandonar este proyecto pendiente.

Resumen

Este proyecto de Ingeniería Eléctrica Electrónica en Telecomunicaciones consiste en la implementación de un sistema electrónico de alarma e inmovilización vehicular tecnológicamente innovador ya que emplea tecnología Bluetooth y el uso de smartphone. El propósito de este trabajo reside en la sustitución del dispositivo de radiofrecuencia “Control Remoto tradicional” por una aplicación instalada en el smartphone del dueño del vehículo.

Esta tesis se desarrolló en tres segmentos. La creación de una aplicación smartphone intuitiva e invulnerable; el establecimiento de una conexión Bluetooth autónoma e indescifrable entre el smartphone y el microcontrolador; y, la implementación de un microcontrolador y un circuito electrónico que sea multitarea, de bajo consumo eléctrico, compacto y resistente a factores climáticos.

Una vez concluido el proceso anteriormente descrito, los resultados se integran para verificar si los objetivos planteados al inicio de esta investigación fueron alcanzados con satisfacción.

El hardware y software utilizados en esta tesis fueron: el IDE Microsoft Visual Studio 2008, el *smartphone* Samsung i637 con Windows Mobile 6.5, el módulo Bluetooth RN-41, el microcontrolador ATMEGA8, el compilador Bascom, el simulador Proteus y un circuito electrónico profesional.

Abstract

This project of Electrical-Electronic Engineering in Telecommunications consists in the implementation of an electronic alarm and vehicle immobilizer technologically innovative because it employs Bluetooth technology and the use of smartphone. The purpose of this work lies in the replacement of the radio frequency device "traditional Remote Control" for an application installed on the smartphone of the vehicle´s owner.

This thesis was developed in three segments. The creation of an intuitive and invulnerable smartphone application; the establishing of an autonomous and indecipherable Bluetooth connection between the smartphone and the microcontroller; and, the implementation of a microcontroller and an electronic circuit that is multitasking, low power consuming, compact and resistant to climatic factors.
Once completed the process described above, the results are integrated to check if the objectives set at the beginning of this research were achieved with satisfaction.

The hardware and software used in this thesis were: Microsoft Visual Studio 2008 IDE, the Samsung i637 smartphone with Windows Mobile 6.5, the RN-41 Bluetooth module, the ATMEGA8 microcontroller, the compiler Bascom, Proteus simulator and a professional electronic circuit.

Tabla de Contenido

DEDICATORIA .... **V**
AGRADECIMIENTOS .... **VI**
RESUMEN .... **VII**
ABSTRACT .... **VIII**
TABLA DE CONTENIDO .... **IX**
LISTA DE FIGURAS .... **XII**
LISTA DE TABLAS .... **XIV**
LISTA DE ECUACIONES .... **XV**
LISTA DE VIDEOS .... **XVI**

CAPÍTULO 1 .... **1**
1. INTRODUCCIÓN **1**
1.1 ANTECEDENTES 1
1.2 JUSTIFICACIÓN DEL PROYECTO 3
1.3 PROPUESTA Y OBJETIVOS 3
1.4 CONTENIDO 6

CAPÍTULO 2 .... **8**
2. FUNDAMENTOS TEÓRICOS **8**
2.1 SISTEMA DE ENCENDIDO VEHICULAR 8
2.1.1 Conexiones cilindro de arranque 11
2.2 ENTORNO INTEGRADO DE DESARROLLO MICROSOFT VISUAL STUDIO 2008 12
2.3 SAMSUNG I637 JACK 14
2.4 BLUETOOTH 16
2.4.1 Reseña histórica 16
2.4.2 Especificaciones desarrolladas 16
2.5 CLASS 1 BLUETOOTH MODULE RN-41 18
2.6 MODELO DE COMUNICACIÓN USART, UART, TTL, RS-232 20
2.6.1 Comunicación USART 20
2.6.2 Conexión UART 21
2.6.3 Conexión TTL 22
2.6.4 Conexión RS-232 23
2.7 MICROCONTROLADOR AVR, ATMEGA8 24
2.8 COMPILADOR BASCOM Y SIMULADOR PROTEUS 27
2.8.1 Bascom-AVR 27
2.8.2 Proteus VSM 29

2.9 CIRCUITO ELECTRÓNICO DE CONTROL ... 30
2.9.1 Circuito Integrado, regulador de voltaje ... 30
2.9.2 Transistores bipolares ... 32
2.9.3 Relés electromagnéticos ... 34

CAPÍTULO 3 ... 36
3. METODOLOGÍA ... 36
3.1 PROGRAMACIÓN EN VISUAL STUDIO ... 36
3.1.1 Subpanel "Off Engine" ... 37
3.1.2 Subpanel "On Engine" ... 38
3.1.3 Subpanel "Accessories" ... 39
3.1.4 Subpanel "Doors" ... 40
3.1.5 Subpanel "Password Setting" ... 41
3.1.6 Subpanel "Car Keys Settings" ... 42
3.2 PROPIEDADES Y CONFIGURACIÓN DEL PUERTO SERIAL ... 43
3.3 PROCESO PARA EJECUTAR LA APLICACIÓN INTERFACE GRÁFICA DE USUARIO EN EL *SMARTPHONE* 46
3.4 PROGRAMACIÓN MÓDULO BLUETOOTH RN-41 ... 49
3.5 PROGRAMACIÓN ATMEGA8 ... 51
3.5.1 Subrutinas "Off Engine" ... 52
3.5.2 Subrutina "On Engine" ... 53
3.5.3 Subrutinas "Open & Lock Doors", "w/ or w/o keys" y "Accessories" ... 55
3.6 SIMULACIONES EN PROTEUS ... 56
3.7 GRABACIÓN DEL MICROCONTROLADOR CON "PROGISP" ... 60
3.8 DISEÑO DEL CIRCUITO IMPRESO ... 61
3.8.1 Divisores de voltaje y dispositivos de seguridad electrónicos ... 62
3.8.2 Circuito electrónico, módulo Bluetooth ... 63
3.8.3 Circuito electrónico, microcontrolador, puertos de entrada y salida ... 63
3.8.4 Arreglo de transistores NPN y diodos emisores de luz ... 64
3.8.5 Arreglo de relés electromagnéticos ... 67
3.9 RELÉS ELECTROMAGNÉTICOS PARA EL CIRCUITO ELÉCTRICO ... 69
3.10 CIRCUITO ELECTRÓNICO DE PRUEBA ... 70
3.11 ADECUACIONES PARA EL SISTEMA DE ALARMA E INMOVILIZACIÓN VEHICULAR ... 72

CAPÍTULO 4 ... 75
4. PRUEBAS Y RESULTADOS ... 75
4.1 PRUEBAS Y RESULTADOS EN LA IMPLEMENTACIÓN DEL CIRCUITO ELECTRÓNICO DE CONTROL ... 75
4.2 PRUEBAS Y RESULTADOS SOBRE LA INTERFACE GRÁFICA DE USUARIO EN EL *SMARTPHONE* ... 80

4.3 PRUEBAS Y RESULTADOS EN LA CONEXIÓN ENTRE EL SMARTPHONE Y EL MÓDULO BLUETOOTH . 83

CAPÍTULO 5 86

5. CONCLUSIONES Y RECOMENDACIONES 86

5.1 CONCLUSIONES 86

5.2 RECOMENDACIONES Y EXPECTATIVAS COMERCIALES 89

BIBLIOGRAFÍA 91

ANEXOS 95

CÓDIGO IMPLEMENTADO EN EL MICROCONTROLADOR ATMEGA8 95

CÓDIGO IMPLEMENTADO EN LA APLICACIÓN *SMARTPHONE* 99

Lista de Figuras

Figura 1. Partes del sistema de alarma de un vehículo. (Harris 1) 2
Figura 2. Propuesta y Objetivos del Proyecto de Tesis. 4
Figura 3. Funcionamiento del sistema de encendido por bobina. (Read and Reid 61) 8
Figura 4. Sistema de encendido. Posición “Start”. (Billiet 499) 10
Figura 5. Sistema de encendido. Posición “Ignición”. (Billiet 499) 11
Figura 6. Conexión de cables en cilindro de arranque 12
Figura 7. Selección de plantillas, “Smart Devices” en Visual Studio 2008. 13
Figura 8. Ejemplo de emulador Pocket PC 2003. 14
Figura 9. Samsung i637 Jack. (st2.gsmarena 1) 15
Figura 10. Rango de frecuencia de Bluetooth y WiFi. (Krenik y Yonker 1) 17
Figura 11. Módulo Bluetooth RN-41. (Roving Networks , RN-41 1) 18
Figura 12. Transmisión sincrónica. (uControl 10) 21
Figura 13. Transmisión asincrónica. (TextosCientificos párr. 1) 22
Figura 14. Comparación transmisión asincrónica, TTL y RS-232. (Cheng Yu párr. 1) 23
Figura 15. Microcontrolador ATMEGA8. (Sparkfun párr. 3) 24
Figura 16. Software BASCOM-AVR. 29
Figura 17. Software Proteus VSM. 30
Figura 18. Características de un Regulador de Voltaje. (Semtech Electronics Ltd. 1) 30
Figura 19. Circuito típico de aplicación de un Regulador de Voltaje. (Cisilino 73) 31
Figura 20. Características Transistor N-P-N. (Semtech Electronics Ltd. 1) 32
Figura 21. Modelo simple de transistor N-P-N como una válvula. (Arrayás 281) 33
Figura 22. Manera en cómo un ohmímetro ve un transistor N-P-N. (Arrayás 282) 34
Figura 23. Relé no excitado. (Cisilino 60) 35
Figura 24. Relé excitado. (Cisilino 61) 35
Figura 25. Diagrama de flujo Visual Studio, subpanel “Main Menu”. 36
Figura 26. Diagrama de flujo, Subpanel “Off Engine” 37
Figura 27. Diagrama de flujo, subpanel “On Engine” 38
Figura 28. Diagrama de flujo, subpanel “Accessories”. 39
Figura 29. Diagrama de flujo, subpanel “Doors” 40
Figura 30. Diagrama de flujo, subpanel “Password Setting” 41
Figura 31. Diagrama de flujo, subpanel “Car Keys Settings”. 42
Figura 32. Pantalla “Bluetooth Devices”, opción “COM Ports”. 43
Figura 33. Pantalla “COM Ports”, opción “New Outgoing Port” 44
Figura 34. Pantalla “COM Ports”, opción COM2. 44
Figura 35. Configuración de puerto de comunicación en Visual Studio. 45
Figura 36. Propiedades del puerto serial en Visual Studio. 46

Figura 37. Proceso para ejecutar la aplicación en el *smartphone*, Paso 1. ... 47
Figura 38. Proceso para ejecutar la aplicación en el *smartphone*, Paso 2. ... 47
Figura 39. Aplicación ejecutándose sobre el *smartphone.* ... 48
Figura 40. Carpeta donde se encuentra el archivo ejecutable de la aplicación. ... 48
Figura 41. Conexión física para programar módulo Bluetooth. ... 49
Figura 42. Propiedades HyperTerminal para conectarse módulo Bluetooth. ... 49
Figura 43. HyperTerminal y confirmación de conexión exitosa con módulo Bluetooth. ... 50
Figura 44. Diagrama de flujo microcontrolador, "Main Menu". ... 51
Figura 45. Subrutinas "Off Engine". ... 52
Figura 46. Subrutinas "On Engine". ... 53
Figura 47. Subrutinas "Open & Lock Doors", "w/ or w/o keys" y "Accessories". ... 55
Figura 48. Simulaciones del µC sobre ISIS de Proteus. ... 57
Figura 49. Microcontrolador y circuito de grabación "PROGISP". ... 60
Figura 50. Grabación microcontrolador con "PROGISP". ... 61
Figura 51. Diagrama del esquema del circuito electrónico. ... 62
Figura 52. Diagrama Transistor, calculo Rb. ... 65
Figura 53. Simulación sobre el transistor N2222A. ... 66
Figura 54. Conexión transistor N2222A con LED y relé. ... 67
Figura 55. Diagrama de conexiones relés y funciones del automóvil. ... 68
Figura 56. Circuito electrónico de control instalado sobre *protoboard.* ... 71
Figura 57. Circuito electrónico de control y el arreglo de relés. ... 71
Figura 58. Conexión de cables: Circuito Primario y cilindro del vehículo. ... 73
Figura 59. Diodo de protección en el botón del vehículo. ... 73
Figura 60. Servomotor instalado en la puerta del vehículo. ... 73
Figura 61. Conexión en paralelo al bulbo del freno. ... 74
Figura 62. Conexión en paralelo al relé de la bocina del vehículo. ... 74
Figura 63. Pruebas realizadas sobre Mini Austin con el circuito electrónico de prueba. ... 76
Figura 64. Layout circuito de electrónico de control. ... 77
Figura 65. Layout circuito electrónico de control, arreglo de relés. ... 78
Figura 66. Circuito electrónico de control final extendido. ... 79
Figura 67. Circuito electrónico de control final compacto. ... 79
Figura 68. Ubicación del circuito de control en el Mini Austin. ... 80

Lista de Tablas

TABLA 1. POSICIONES DEL CILINDRO DE ARRANQUE. 12
TABLA 2. ESPECIFICACIONES MÓDULO BLUETOOTH RN-41. (ROVING NETWORKS , DATASHEET 1)......... 19
TABLA 3. ESPECIFICACIONES DE MICROCONTROLADOR ATMEGA8. (ATMEL CORPORATION 1) 26
TABLA 4. REGULADORES DE TENSIÓN POSITIVA. (TENSIÓN EXPRESADAS EN VOLTIOS). (CISILINO 74) 32
TABLA 5. PARÁMETROS DE CONFIGURACIÓN DE MÓDULO BLUETOOTH... 50
TABLA 6. RESULTADOS DE LA SIMULACIÓN DE SUBRUTINAS EN MICROCONTROLADOR............................. 58
TABLA 7. SUBPANELES DE LA APLICACIÓN INSTALADA EN EL *SMARTPHONE.* 81
TABLA 8. FUNCIONES ESPECIALES DE LA APLICACIÓN INSTALADA EN EL *SMARTPHONE.*........................... 82

Lista de Ecuaciones

ECUACIÓN 1. FACTOR DE GANANCIA DEL TRANSISTOR. 64
ECUACIÓN 2. NUEVA CORRIENTE BASE I_B PARA SATURAR EL TRANSISTOR. 65
ECUACIÓN 3. CÁLCULO DE R_B. 65

Lista de Videos

VIDEO 1. SUBRUTINA "OFFENGINE" ACTIVADA POR EL PIN DE ENTRADA DEL µC. 58
VIDEO 2. SUBRUTINA "ONENGINE" ACTIVADA POR EL PIN DE ENTRADA DEL µC. 58
VIDEO 3. SUBRUTINA "PRESTAR Y STARTENGINE" ACTIVADA POR EL PIN DE ENTRADA DEL µC. 58
VIDEO 4. SUBRUTINA "OFFENGINE" ACTIVADA A TRAVÉS DEL PUERTO SERIAL DEL µC. 59
VIDEO 5. SUBRUTINA "ONENGINE" ACTIVADA A TRAVÉS DEL PUERTO SERIAL DEL µC. 59
VIDEO 6. SUBRUTINA "OPEN-CLOSE DOOR" ACTIVADA A TRAVÉS DEL PUERTO SERIAL DEL µC. () 59
VIDEO 7. SUBRUTINA "W/ OR W/O KEYS" ACTIVADA A TRAVÉS DEL PUERTO SERIAL DEL µC. 59
VIDEO 8. SUBRUTINA "ACCESSORIES" ACTIVADA A TRAVÉS DEL PUERTO SERIAL DEL µC. 59
VIDEO 9. PRUEBA DE ENCENDIDO DEL VEHÍCULO A TRAVÉS DEL BOTÓN "ONENGINE" 76
VIDEO 10. FUNCIÓN "OFF ENGINE" DE LA APLICACIÓN VISUAL STUDIO. 81
VIDEO 11. FUNCIÓN "ON ENGINE" DE LA APLICACIÓN VISUAL STUDIO. 81
VIDEO 12. FUNCIÓN "ACCESSORIES" DE LA APLICACIÓN VISUAL STUDIO. 82
VIDEO 13. FUNCIÓN "OPEN" & "LOCK" DOOR DE LA APLICACIÓN VISUAL STUDIO. 82
VIDEO 14. SUBPANEL "PASSWORD SETTING" DE LA APLICACIÓN VISUAL STUDIO. 82
VIDEO 15. FUNCIONES "W/ & W/O KEYS" DE LA APLICACIÓN VISUAL STUDIO. 82
VIDEO 16. FUNCIÓN ESPECIAL, CONEXIÓN BLUETOOTH ESTABLECIDA PARA ENCENDER EL VEHÍCULO...... 83
VIDEO 17. FUNCIÓN ESPECIAL, INFORMAR LA ACCIÓN DE PRESIONAR EL PEDAL DEL FRENO. 83
VIDEO 18. FUNCIÓN ESPECIAL, CIFRADO DE CLAVES. 83
VIDEO 19. PRUEBA PARA VERIFICAR QUE LA CONEXIÓN BLUETOOTH ES AUTÓNOMA. 84
VIDEO 20. PRUEBA PARA VERIFICAR *FEATURE BONDING* EN EL MÓDULO RN-41. 84
VIDEO 21. PRUEBA PARA VERIFICAR *FEATURE INVISBLE* EN EL MÓDULO RN-41. 84

CAPÍTULO 1

1. Introducción

El primer caso documentado sobre un automóvil robado fue en 1896, tan solo una década después del lanzamiento del primer vehículo impulsado a gasolina. Desde su introducción hasta el día de hoy, los automóviles han sido objetivo de atención para los ladrones: son valiosos, razonablemente fáciles de vender y tienen el sistema fugitivo incorporado. Algunos estudios reportan que cinco a seis vehículos son robados cada día en la ciudad de Quito durante el año 2012 (El Comercio , párr. 3). A la luz de esta alarmante estadística, no es de extrañarse que miles de ecuatorianos inviertan y compren alarmas vehiculares equipadas con sofisticados sensores electrónicos, sirenas a todo volumen y sistemas de activación remota. (Harris)

1.1 Antecedentes

En el mercado ecuatoriano de auto-lujos y accesorios vehiculares, se puede adquirir alarmas para autos que se componen principalmente de uno o varios sensores conectados a una especie de sirena; sin embargo, existen otros sistemas más sofisticados que se componen de una unidad de control llamada “Cerebro” y un emisor de radiofrecuencia llamado “Control Remoto”, el cual controla inalámbricamente el “Cerebro” antes mencionado (Ver Figura 1).

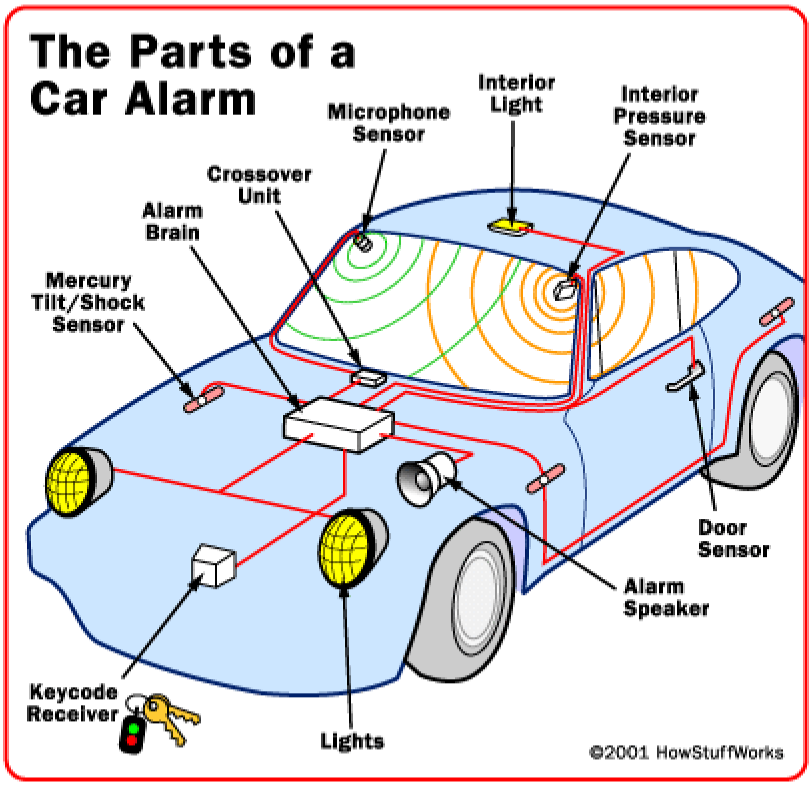


Figura 1. Partes del sistema de alarma de un vehículo. (Harris 1)

El "Control Remoto" de las alarmas vehiculares ha mejorado cualitativamente en estos últimos años, sus primeros ejemplares eran dispositivos de infrarrojo que solamente podían hacer una función, sin embargo, con el avance de la tecnología han pasado a ser interfaces inteligentes conocidas como "Interfaces Inteligentes de Autos" (Abu-Ein, Abu-Ein y Fayyad 3273). Formando parte de esta evolución inalámbrica, este proyecto de tesis desea implementar una nueva e innovadora solución de comunicación inalámbrica en alarma vehicular, distinta a las soluciones existentes actualmente en el país. Dicha solución será descrita en la sección 1.3.

1.2 Justificación del proyecto

Las investigaciones de Brad A. Myers concluyeron que los dispositivos móviles pueden ser compatibles entre sí. Esto llevó a una serie de interesantes y nuevos temas de investigación, como por ejemplo: Si un usuario posee un dispositivo móvil con una pantalla de alta definición y un método de tecleado adecuado, ¿por qué un “Control Remoto” de baja calidad es usado para una aplicación? Sus estudios preliminares sugieren que las personas pueden utilizar un “Control Remoto” en su *PDA,* a la mitad del tiempo y con la mitad de errores a comparación de interfaces de otros fabricantes. (Abu-Ein, Abu-Ein y Fayyad 3274)

Myers en su enunciado anterior corrobora la justificación e importancia de esta tesis; desarrollar una aplicación de control inalámbrico en un *smartphone*, con el objetivo de actualizar y reemplazar al “Control Remoto” que existente actualmente en el mercado de las alarmas vehiculares. Los detalles sobre esta justificación serán descritos en la siguiente sección.

1.3 Propuesta y Objetivos

La propuesta de este proyecto está enfocada en actualizar los componentes de un sistema de alarma comercial, con tecnología innovadora, pero a la vez que sea asequible para un estudiante de pregrado.

Esta idea de implementar un “Control Remoto” a través de una aplicación programada en un teléfono inteligente, aparece por dos necesidades particulares: instalar una alarma en un Mini Austin Morris restaurado y presentar un proyecto

de grado para la obtención del título de Ingeniería Eléctrica Electrónica especialidad de Telecomunicaciones. Estos motivos plantean el siguiente objetivo de tesis: diseñar e implementar un sistema de alarma e inmovilizador vehicular controlado por un *smartphone*, a través de comunicación Bluetooth.

Los objetivos específicos se pueden inferir a partir del objetivo principal propuesto; sin embargo, para una mejor comprensión de los mismos, el proyecto se ha dividido en tres partes principales (ver Figura 2).

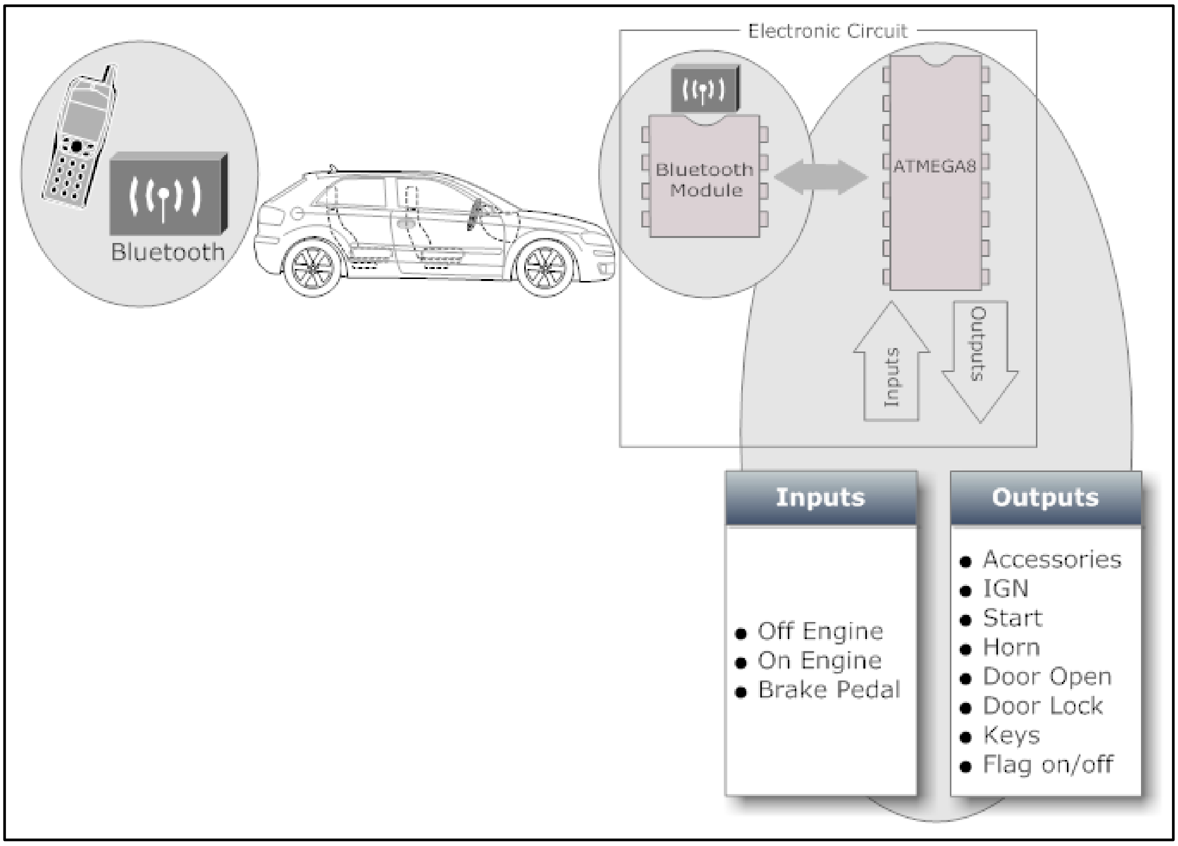


Figura 2. Propuesta y Objetivos del Proyecto de Tesis.

1) Crear una aplicación para *smartphone* con las siguientes características:

- Ser intuitiva, para que el usuario final pueda controlar el sistema de alarma satisfactoriamente.
- Ser invulnerable, para que en caso que el smartphone sea robado, el ladrón no pueda acceder a la misma y así, impedir una posible sustracción parcial o total del vehículo.

2) Instalar un “Módulo Bluetooth” en el vehículo para levantar una conexión entre el *smartphone* y el microcontrolador ATMEGA8. El enlace mencionado debe tener las siguientes características:

- Ser autónomo, para que el dueño no tenga que preocuparse si la conexión está levantada o no.
- Ser indescifrable, implementar un método de seguridad que no permita a extraños *hackear* el “Módulo Bluetooth”.

3) Implementar un microcontrolador y un circuito electrónico con las siguientes características:

- Ser multitarea, el microcontrolador debe tener la suficiente capacidad para ejecutar varios procesos de entrada como de salida.
- Ser de bajo consumo eléctrico, cuando el vehículo esté apagado, el consumo eléctrico del circuito electrónico debe ser lo menor posible para evitar descargar la batería del mismo.
- Ser compacto, a pesar de ejecutar varias tareas, el circuito debe ser lo suficientemente pequeño para poder instalarse en un sitio escondido del Mini Austin.
- Ser resistente a factores climáticos, los factores de humedad, temperatura y polvo pueden afectar el rendimiento del circuito.

1.4 Contenido

El primer capítulo presenta los antecedentes, la justificación, la propuesta, y los objetivos planteados para este proyecto de tesis. Los antecedentes es un extracto sobre la invención de las alarmas vehiculares y sus dispositivos de radiofrecuencia (RF). La justificación consiste en el desarrollo de una aplicación de control inalámbrica innovadora que actualice y reemplace al “Control Remoto tradicional”. La propuesta se plantea en función de dos necesidades personales; instalar una alarma vehicular en un Mini Austin Morris restaurado y presentar una tesis para la obtención del título universitario. Para lograr dicho cometido se hace una síntesis de los tres segmentos en que se divide el proyecto y sus respectivos objetivos a alcanzar.

El segundo capítulo presenta los fundamentos teóricos de las diversas partes que conforman el software y el hardware del sistema a implementarse. Los principales temas que sintetizan este marco teórico son; el sistema de encendido vehicular, el entorno de programación Visual Basic 2008, el smartphone Samsung i637, el protocolo de comunicación Bluetooth, el módulo Bluetooth RN-41, el modelo de comunicación USART, TTL y RS-232, el microcontrolador Atmega8, el compilador Bascom, el simulador Proteus, y el circuito electrónico de control y sus partes principales.

El tercer capítulo es la metodología del proyecto. Se describe paso a paso cómo se diseña y se implementa los tres segmentos planteados en esta tesis, de tal manera, que pueda ser reproducida posteriormente por cualquier persona que esté interesada en hacerlo.

El capítulo cuatro es la presentación total del trabajo, se realizan pruebas y verificaciones sobre el mismo para obtener y analizar sus resultados, previo al desarrollo de las conclusiones respectivas.

El capítulo cinco es la presentación de las conclusiones del proyecto, las recomendaciones para trabajos futuros y las expectativas comerciales que se tiene sobre el mismo.

CAPÍTULO 2

2. Fundamentos Teóricos

2.1 Sistema de encendido vehicular

Uno de los objetivos de este proyecto de tesis, es llegar a ser implementada en un vehículo real. Con esta situación, el dueño tiene la obligación de conocer, al menos de forma general, los componentes electrónicos que serán controlados por su s*martphone*. A continuación se describe a manera conceptual y resumida, el sistema de encendido vehicular.

De forma gráfica se presenta los componentes básicos de un sistema típico de arranque vehicular (ver Figura 2).

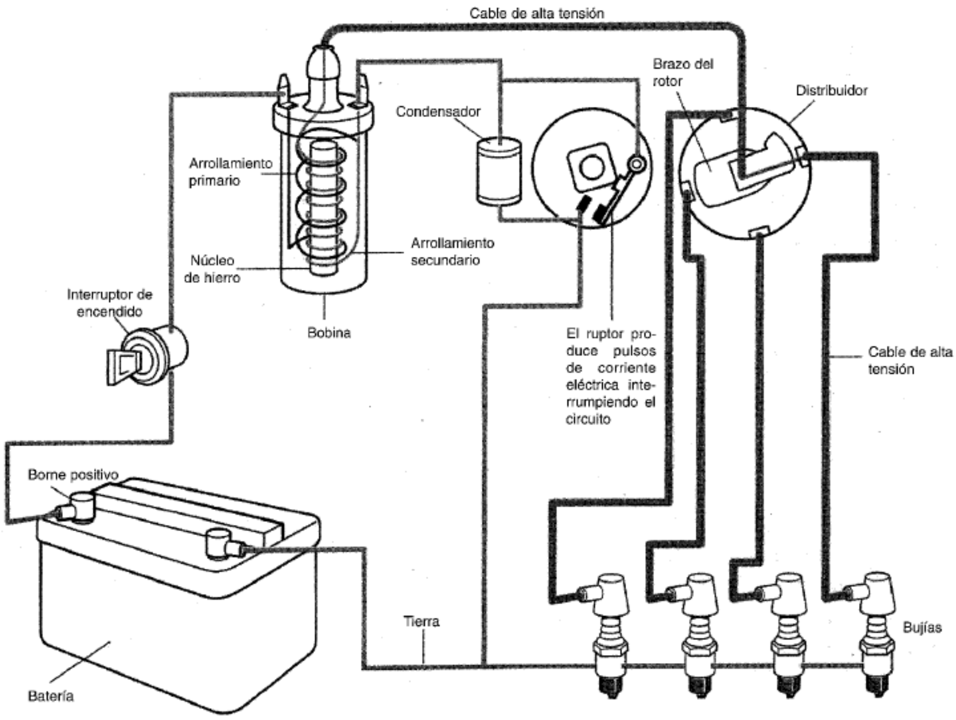


Figura 3. Funcionamiento del sistema de encendido por bobina. (Read and Reid 61)

La **batería** suministra electricidad.

El **interruptor de encendido** controla el sistema (y actúa como un sistema antirrobo) permitiendo o no, el paso de corriente desde la batería.

La **bobina** transforma la baja tensión en alta tensión necesaria para generar en las bujías la chispa que quema la mezcla.

Funcionando a media velocidad, el **distribuidor** distribuye la corriente de alta tensión de la bobina entre cada bujía por turnos, en el momento preciso en que es requerida.

Los cables de **alta tensión (AT)** y de **baja tensión (BT)** se diferencian en que los cables BT son finos y los AT son mucho más gruesos. Esto nos hace ver que el sistema de encendido será constituido por dos circuitos.

La BT procedente de la batería necesita relativamente bajo aislamiento. Normalmente funcionan sólo a 12V y no se corren graves riesgos de electrocución con ella. Esto se conoce como **Circuito Primario**.

Los cables de AT comunican la bobina, el distribuidor y las bujías. Es necesario un gran aislamiento para prevenir fugas de corriente que pueden provocar electrocuciones de unos 20.000 voltios si se toca en el lugar inadecuado. Otro nombre para esta parte del sistema es el **Circuito Secundario**. (Read y Reid 60-62)

Cuando el interruptor de encendido se coloca en la posición “ST” o “Start”, (ver Figura 4), se establece una conexión entre la batería y el borne positivo de la bobina, a la vez que el motor de arranque comienza a girar el cigüeñal. Mientras dura el periodo de arranque, la bobina es alimentada por la totalidad de la tensión disponible en la batería.

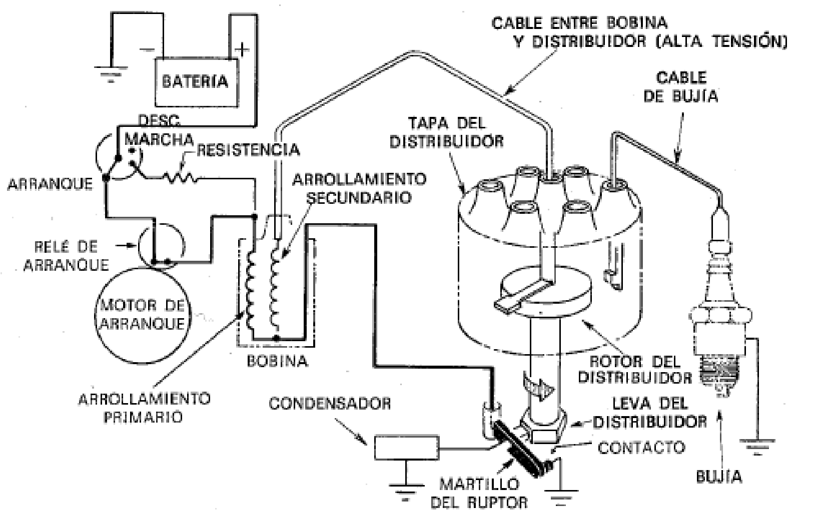


FIG. 12-11. Trayectoria de la corriente eléctrica primaria durante el período de arranque. Mientras el interruptor de encendido se encuentra en la posición de «arranque», llega corriente al motor de arranque, la resistencia del primario queda en derivación y el resto del circuito primario recibe la corriente directamente desde el motor de arranque.

Figura 4. Sistema de encendido. Posición “Start”. (Billiet 499)

Finalizado el arranque, el interruptor de arranque retorna a posición de “ON”, (ver Figura 5), la resistencia se intercala en el circuito, limitando el valor máximo de la corriente primaria que atraviesa la bobina y el ruptor. El motor de arranque se desactiva. (Billiet 498 - 450)

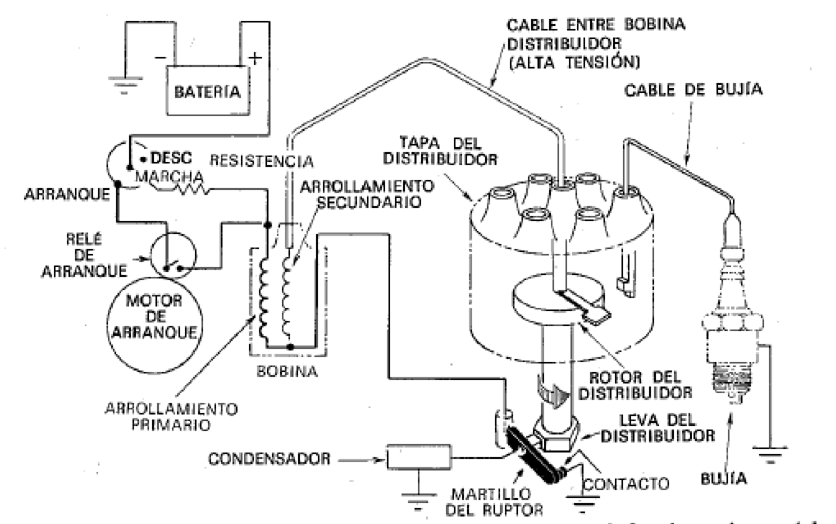


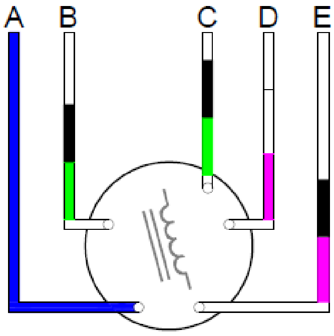
FIG. 12-12. **Trayectoria de la corriente primaria durante el funcionamiento del motor.** El interruptor de arranque se encuentra en la posición de «marcha». En este caso, la corriente atraviesa la resistencia del primario antes de entrar en el resto del circuito primario y regresar por masa a través de los contactos del distribuidor cuando éstos se cierran. En la bobina se crea campo magnético.

Figura 5. Sistema de encendido. Posición "Ignición". (Billiet 499)

2.1.1 Conexiones cilindro de arranque

El cilindro de arranque es una cerradura que está integrada al sistema de encendido o arranque del vehículo. Para activar y desactivar el seguro del cilindro, se necesita insertar una llave específica. El cilindro posee dos funciones principales, brindar seguridad para evitar encender el automóvil sin autorización del dueño y controlar el **Circuito Primario** de encendido del vehículo.

La conexión de los cables del sistema eléctrico de encendido del Mini Austin, se muestra a continuación (ver Figura 6).



Cylinder Connection			
A	ACC	Accessories	blue
B	IG1	Ignition 1	black-green
C	IG2	Ignition 2	back-white
D	B	Battery	white-pink
E	ST	Start	black-pink

Figura 6. Conexión de cables en cilindro de arranque.

El cilindro posee cuatro posiciones. Al girarlo con la llave, conecta y desconecta los cables del sistema de encendido del vehículo. A continuación se muestra las posiciones y las conexiones que los cables adquieren al girar la llave (ver Tabla 1).

Posición #	Nombre	Configuración
- Posición # 0	"OFF"	Cables Aislados
- Posición # 1	"ACC"	Batería ACC
- Posición # 2	"ON"	Batería IGN ACC
- Posición # 3	"START"	Batería ST IGN

Tabla 1. Posiciones del cilindro de arranque.

2.2 Entorno Integrado de Desarrollo Microsoft Visual Studio 2008

Microsoft Visual Studio es un Entorno Integrado de Desarrollo, también conocido como IDE por sus siglas en ingles (Integrated Development Environment). Se ha desarrollado para la creación de software, aplicaciones, servicios web en

cualquier entorno que soporte la plataforma ".NET". Adicionalmente, con la versión 2008, se puede crear aplicaciones que intercomuniquen estaciones de trabajo, páginas web y dispositivos móviles. (Willydev 5).

Microsoft Visual Studio 2008 permite desarrollar aplicaciones de usuario que posteriormente pueden ser ejecutadas en dispositivos móviles como *Smartphones* cuya plataforma sea Windows CE o Windows Mobile. Se puede utilizar Visual C#, Visual Basic o Visual C++ para escribir aplicaciones administrativas que se ejecutan en ".NET Compact Framework". Para cualquier lenguaje de programación que se escoja, el diseño de Interface Gráfica de Usuario (GUI), el Depurador y el Editor de Código es el mismo que se utiliza cuando se desarrollan aplicaciones para un ordenador. Sin embargo, para el caso de dispositivos móviles, se selecciona una de las plantillas de proyecto "Smart Devices" (ver Figura 7), disponibles para el lenguaje de programación Visual Basic Studio 2008 y Visual Basic Studio 2010 (Microsoft , párr. 1)

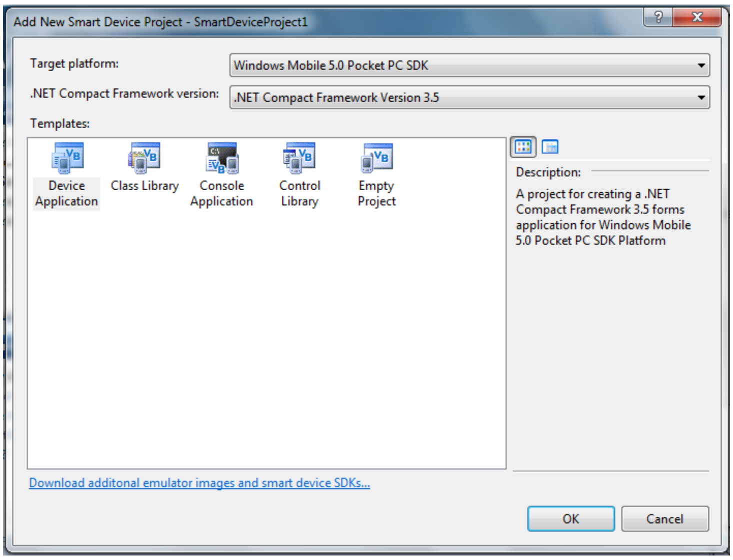


Figura 7. Selección de plantillas, "Smart Devices" en Visual Studio 2008.

Visual Studio 2008 proporciona emuladores (ver Figura 8), que permiten ejecutar y depurar código antes de su implementación final en el dispositivo móvil del usuario. La simulación se logra mediante herramientas que simplifican el empaquetado de los recursos y de la aplicación en archivos *.CAB. (Microsoft , párr. 2)



Figura 8. Ejemplo de emulador Pocket PC 2003.

Los emuladores antes mencionados, se utilizarán para hacer pruebas y emulaciones con la finalidad de depurar la Interface Gráfica Usuario (IGU) a implementarse en el *smartphone* Samsung i637.

2.3 Samsung i637 Jack

El Samsung i637 Jack es un *smartphone* con teclado QWERTY y una pantalla horizontal QVGA de 2.4 pulgadas (ver Figura 9). Corre sobre Windows Mobile 6.1 estándar pero puede ser actualizado a la versión 6.5 una vez que sea lanzado al mercado. Posee A-GPS integrado, cámara de 3 mega-pixeles, y soporta tribanda UMTS/HSDPA. Está equipado con Bluetooth A2DP, tecnología que permite el

consumo bajo de energía o batería en la transmisión de datos sin cables entre otros dispositivos tecnológicos (smart-GSM , párr.1).



Figura 9. Samsung i637 Jack. (st2.gsmarena 1)

Mike McCamon, director ejecutivo de “Bluetooth Special Interest Group” (SIG), manifiesta “no hay tecnología que pueda hacer lo que el Bluetooth puede hacer”, apuntando a que el Bluetooth es bien conocido por su conexión de corto alcance, de bajo consumo de energía en interfaces pequeñas y económicas. (Motsay ,8)

McCamon confirma que Bluetooth es un sistema de interfaz, que permite que dispositivos dispersos, se comuniquen. Con ello se alcanza otro de los objetivos para la implementación de la tesis; utilizar una interfaz inalámbrica para enlazar comunicación entre el *smartphone* y el microcontrolador, a través de un módulo Bluetooth instalado en el circuito electrónico de control.

2.4 Bluetooth

2.4.1 Reseña histórica

A principios de 1994, un equipo de investigadores dirigido por Sven Mattisson y Jaap Haartsen en la ciudad universitaria de Lund, Suecia, estaban buscando la posibilidad de desarrollar una conexión inalámbrica entre un audífono de oído y un teléfono. Mientras iban avanzando en la investigación, se dieron cuenta que dicha tecnología excedía las expectativas de solamente crear un audífono inalámbrico, y con ello, por accidente, nació el concepto de Bluetooth (Brooks 14)

2.4.2 Especificaciones desarrolladas

Bluetooth es un sistema de conexión inalámbrica de corto alcance que posibilita la transmisión de voz y datos entre diferentes dispositivos digitales mediante un enlace por radiofrecuencia en la banda de los 2,4 GHz (ver figura 10). Las nuevas versiones y especificaciones de Bluetooth desarrolladas por el Grupo de Interés Especial de Bluetooth (SIG), introdujo nuevas formas de evitar interferencias e impulsar la convivencia entre Bluetooth y otras tecnologías inalámbricas como WiFi; dichas innovaciones son “eSCO channels”, AFH y EDR.

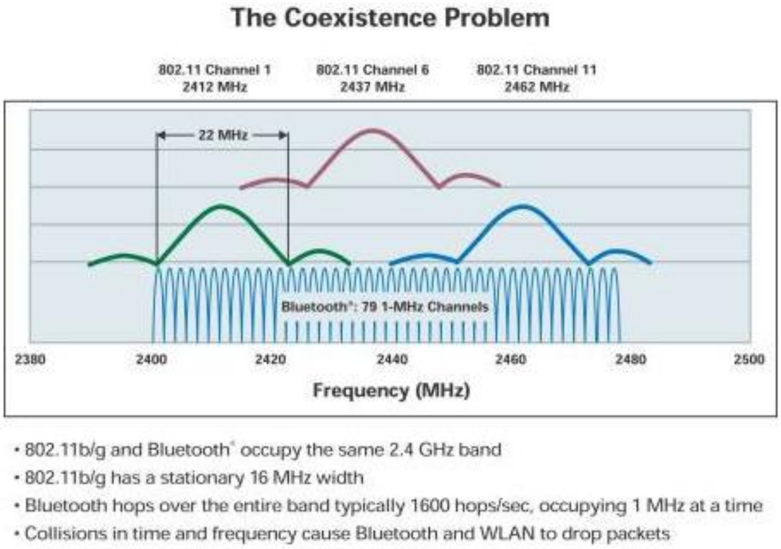


Figura 10. Rango de frecuencia de Bluetooth y WiFi. (Krenik y Yonker 1)

eSCO channels de su acrónimo en inglés (extended synchronous connection oriented), permite que el dispositivo receptor solicite la retransmisión del o de los paquetes perdidos sobre una trama de paquetes recibidos, mejorando así la calidad de la voz en los enlaces de audio.

AFH de su acrónimo en inglés (adaptive frequency hopping), monitorea la calidad del enlace, identifica los canales llenos de las frecuencias de los 2.4 GHz para evitar utilizar dicho canal y frecuencia con la señal de Bluetooth.

EDR de su acrónimo en inglés (Enhanced Data Rate), es una especificación introducida con la versión 2.0 + EDR. Esta funcionalidad provee mayor velocidad de transmisión de datos a una tasa nominal de 3 Mbits por segundo. (Carmody 28)

El constante desarrollo del Bluetooth a través de sus múltiples versiones y especificaciones, han permitido que múltiples dispositivos operen

simultáneamente, incrementando las aplicaciones que se pueden desarrollar para múltiples propósitos investigativos. Para el caso específico de este proyecto de tesis, el objetivo es implementar un módulo Bluetooth, el cual es la interfaz de comunicación entre el teléfono inteligente Samsung i367 y el microcontrolador ATMEGA8.

2.5 Class 1 Bluetooth module RN-41



Figura 11. Módulo Bluetooth RN-41. (Roving Networks , RN-41 1)

El RN-41 es un módulo Bluetooth del tamaño de una moneda de 25 centavos americanos, de bajo consumo eléctrico, y de simple integración con otros dispositivos de diversas marcas y fabricantes (ver Figura 11). A continuación se describe las especificaciones técnicas del módulo Bluetooth (ver Tabla 2).

Bluetooth Versions	2.1 + EDR, 2.0, 1.2, 1.1
Data rate	With onboard stack: 300Kbps: HCI mode: 1.5Mbps sustained, 3Mbps burst
Frequency Band	2.412 - 2.484 GHz
Modulation Techniques	FHSS/GFSK modulation, 79 channels at 1MHz intervals
Profiles	SPP, DUN, HID, iAP, HCI, RFCOM, L2CAP, SDP

Supply voltage	3.3V ± 10%
Output power	+15dBm
Power consumption	Standby/Idle 25 mA : Connected (normal mode) 30 mA : Connected (low power Sniff) 8 mA : Standby/Idle (Deep sleep enabled) 250uA
Operating temperature range	-40C to +85C
Interface	UART, USB, Bluetooth
Antenna	Chip
Size	0.52" x 1" x 0.07" (13.4mm x 25.8mm x 2mm)
Weight	0.055 oz
Certifications	FCC, IC, CE, Bluetooth SIG

Tabla 2. Especificaciones módulo Bluetooth RN-41. (Roving Networks , Datasheet 1)

Las características que permiten que los objetivos del proyecto se cumplan son:

- Versión de Bluetooth 2.0, es la misma versión que posee el Samsung i367. Con esta característica se obtiene la comunicación entre el smartphone y el módulo Bluetooth RN-41.
- Voltaje de funcionamiento 3.3V +/- 10%, éste es el voltaje de polarización del módulo. Es una condición a tomar en cuenta al momento de planificar el circuito electrónico de control que se instalará en el vehículo.
- Características de consumo eléctrico muy bajo, consiguiendo que el circuito electrónico al ser implementado, no descargue la batería del automóvil.
- Consumo de poder en Standby/Idle = 25 mA.
- Conectado (modo normal) = 30 mA.
- Conectado (bajo poder en modo fisgón) = 8 mA.
- Standby/Idle (modo dormido profundo habilitado) = 250uA.

- Temperatura de operación -40°C a +85°C. Los rangos indicados por el fabricante, son temperaturas que existen en el interior del vehículo, lo cual facilita una instalación adecuada del módulo en el interior del mismo.
- **Interfaz UART**, interface que, a través de una conexión TELNET con la laptop se ingresa al “command mode” del módulo para realizar las configuraciones y programaciones del mismo. Adicionalmente, la Interfaz UART es el protocolo que el microcontrolador ATMEGA8 utiliza para comunicarse con el módulo Bluetooth.
- Dimensiones y Peso 13.4mm x 25.8mm x 2mm, 0.055 oz, El módulo tiene un tamaño óptimo comparado a otros módulos que se pueden encontrar en el mercado local.

El módulo Bluetooth RN-41 posee especificaciones que cumplen con los objetivos del proyecto; adquirir un dispositivo de bajo consumo eléctrico para levantar dos tipos de conexiones, una de tipo Bluetooth con el smartphone, y la otra, de tipo serial con el microcontrolador a través del modelo de comunicación USART (UART, TTL, RS-232).

2.6 Modelo de comunicación USART, UART, TTL, RS-232

En esta sección se revisa de manera explícita, los conceptos y fundamentos de la comunicación UART, USART, TTL y RS-232, con el objetivo de comprender mejor las conexiones a utilizarse en el circuito electrónico controlado por el microcontrolador ATMEGA8.

2.6.1 Comunicación USART

USART (Universal Synchronous-Asynchronous Receiver-Transmitter), es tal vez el modelo de comunicación más utilizado entre los módulos de Interfaz Serial. La

principal función del USART es la transmisión o recepción de datos en serie, ya sea de modo sincrónico o asincrónico. Se puede transmitir datos de 8 a 9 bits por trama. Se puede generar interrupciones cuando se produce una recepción de datos o cuando la transmisión ha sido completa. (García Breijo, Compilador C CSS y Simulador PROTEUS para Microcontroladores PIC 168)

La Transmisión Sincrónica (ver Figura 12), utiliza una señal de reloj y una trama de datos; mientras que, en la transmisión asincrónica no se envía la señal de reloj, por lo que el emisor y el receptor deben tener relojes con la misma frecuencia y fase.

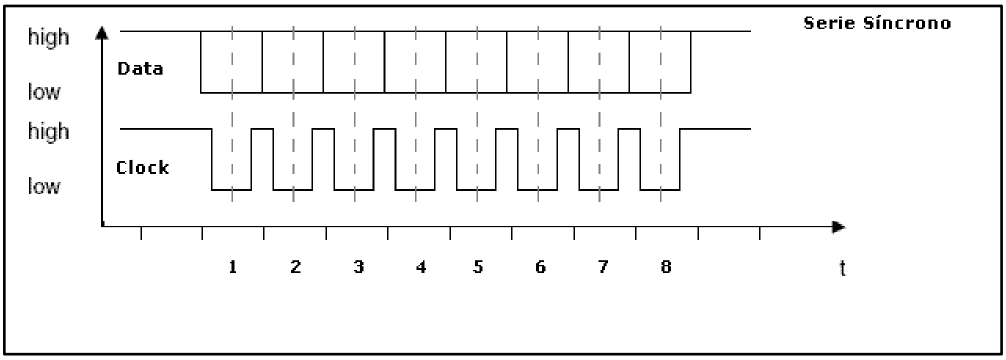


Figura 12. Transmisión sincrónica. (uControl 10)

2.6.2 Conexión UART

UART (Universal Asynchronous Receiver-Transmitter), (ver Figura 13), es un método de comunicación serial asincrónico, cuyo medio físico de transmisión es un cable de cobre o conductor. UART toma bytes de datos y transmite los bits individualmente uno por uno de una manera secuencial. (Osborne 116-117). En el destino, un segundo UART re-ensambla los bits en bytes. Cada UART contiene un registro de desplazamiento o cambio (Shift Register), el cual es el método

fundamental de conversión entre los métodos de comunicación serial y paralelo. (processors.wiki.ti 1)

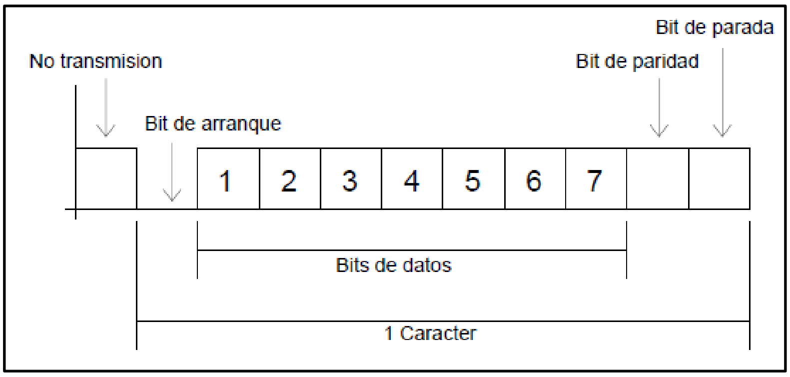


Figura 13. Transmisión asincrónica. (TextosCientificos párr. 1)

2.6.3 Conexión TTL

TTL (Transistor-Transistor Logic), (ver Figura 14), se refiere a un estándar de transmisión asincrónica para circuitos digitales. Fue inventado en la década de 1960 cuando Sylvania y Texas Instruments crearon el modelo TTL basados en circuitos integrados. En los circuitos TTL, hay dos transistores responsables del estado alto o bajo en los pines de salida. Uno de los transistores genera el “cero” lógico, mientras que el otro el “uno” lógico. Con TTL, un “cero” lógico se define como tierra (o por lo general 0 a 0,4 voltios), y “uno” lógico como 5 voltios. (Majon International 1)

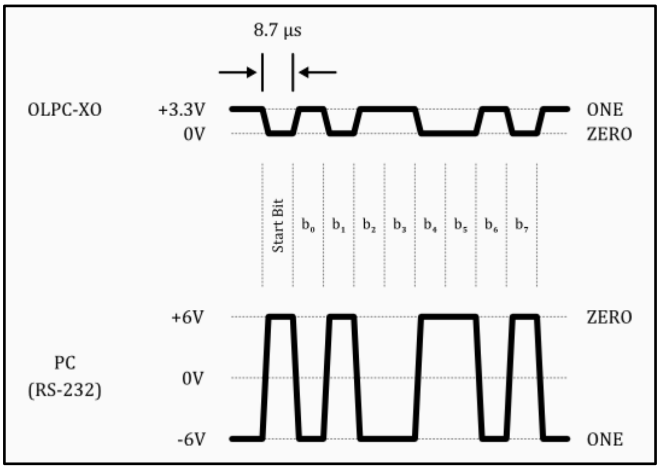


Figura 14. Comparación transmisión asincrónica, TTL y RS-232. (Cheng Yu párr. 1)

2.6.4 Conexión RS-232

RS-232, (ver Figura 14), es un estándar de transmisión asincrónica que define al “cero” lógico como el voltaje mayor que 3 voltios positivo y el “uno” lógico como el voltaje menor que negativo 3 voltios. Voltajes alrededor de cero, que usualmente es tierra, no son considerados una señal en lo absoluto. La pregunta es, ¿por qué usar voltajes positivos y negativos cuando los circuitos integrados generan voltaje o, no voltaje? La razón es porque la degradación de la señal y la interferencia son mucho más probables que se representen en una señal TTL que en una señal RS-232, donde un “uno” es claramente un voltaje negativo y un “cero” es claramente un voltaje positivo. Con este escenario, la configuración TTL se utiliza generalmente en comunicaciones rápidas y cortas dentro de un circuito electrónico, mientras que RS-232 se emplea en conexiones externas y distantes. (Majon International).

2.7 Microcontrolador AVR, ATMEGA8

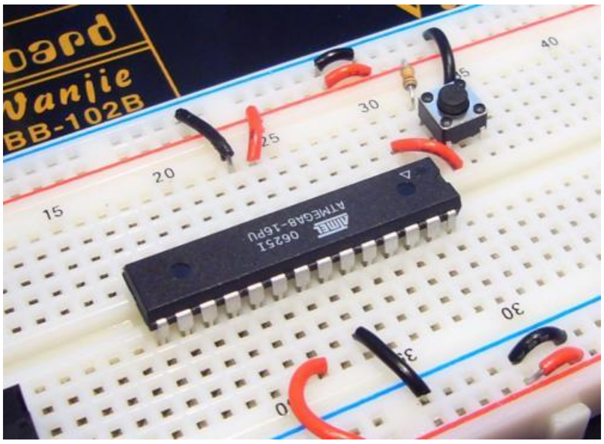


Figura 15. Microcontrolador ATMEGA8. (Sparkfun párr. 3)

A continuación se realiza una breve reseña de las características del microcontrolador que actuaron directamente sobre la planeación e integración del circuito electrónico para esta tesis.

Los fabricantes de automóviles están buscando soluciones para manejar los requerimientos del consumidor sobre la interoperabilidad y la comunicación entre diversas tecnologías dentro del vehículo como son; las aplicaciones de los celulares (redes sociales, mensajería de texto, etc.), con los sistemas del vehículo (indicadores de seguridad, desempeño mecánico, etc.). Para cumplir estos requerimientos, los ingenieros automotrices buscan soluciones de fácil y rápida integración al diseño del vehículo: un peso reducido, de materiales baratos, pero sobre todo, que sean robustos y veloces en procesamiento. Los microcontroladores o los procesadores de 32-bit son la solución. (Anderson 93)

Anderson afirma que, la solución a la interoperabilidad entre las aplicaciones móviles y los sistemas vehiculares, es utilizar microcontroladores o procesadores de 32-bits. Para el objetivo particular de este proyecto de tesis, se va a

implementar el microcontrolador ATMEGA8, cuyas características principales se detallan a continuación (ver tabla 3).

ATMEGA8 High-performance, Low-power Atmel®AVR® 8-bit Microcontroller

• Advanced RISC Architecture
– 130 Powerful Instructions – Most Single-clock Cycle Execution – 32 × 8 General Purpose Working Registers – Fully Static Operation – Up to 16MIPS Throughput at 16MHz – On-chip 2-cycle Multiplier
• High Endurance Non-volatile Memory segments
– 8Kbytes of In-System Self-programmable Flash program memory – 512Bytes EEPROM – 1Kbyte Internal SRAM – Write/Erase Cycles: 10,000 Flash/100,000 EEPROM – Data retention: 20 years at 85°C/100 years at 25°C(1) – Optional Boot Code Section with Independent Lock Bits In-System Programming by On-chip Boot Program True Read-While-Write Operation – Programming Lock for Software Security
• Peripheral Features
– Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode – One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode – Real Time Counter with Separate Oscillator – Three PWM Channels – 8-channel ADC in TQFP and QFN/MLF package Eight Channels 10-bit Accuracy – 6-channel ADC in PDIP package Six Channels 10-bit Accuracy – Byte-oriented Two-wire Serial Interface – Programmable Serial USART – Master/Slave SPI Serial Interface – Programmable Watchdog Timer with Separate On-chip Oscillator – On-chip Analog Comparator
• Special Microcontroller Features
– Power-on Reset and Programmable Brown-out Detection – Internal Calibrated RC Oscillator – External and Internal Interrupt Sources – Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby

• I/O and Packages
– 23 Programmable I/O Lines – 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
• Operating Voltages
– 2.7V - 5.5V (ATmega8L) – 4.5V - 5.5V (ATmega8)
• Speed Grades
– 0 - 8MHz (ATmega8L) – 0 - 16MHz (ATmega8)
• Power Consumption at 4Mhz, 3V, 25°C
– Active: 3.6mA – Idle Mode: 1.0mA – Power-down Mode: 0.5mA

Tabla 3. Especificaciones de microcontrolador ATMEGA8. (Atmel Corporation 1)

Las características que permiten que los objetivos del proyecto se cumplan son:

- Retención de Datos: 20 años a 85°C, 100 años a 25°C; estas características indica lo robusto y duradero que es el microcontrolador a pesar de las temperaturas a las cuales puede exponerse.
- Ejecución de instrucciones en un ciclo de reloj; el ATMEGA8 alcanza rendimientos alrededor de 1 MIPS (Million Instructions Per Second) por MHz, esto permite diseñar un sistema eficiente en relación consumo de corriente vs. velocidad de procesamiento.
- Interfaz Serial de dos cables y programación serial USART; en secciones anteriores se indicó que el Módulo Bluetooth RN-41 posee Interfaz Serial UART, lo cual indicaría que el microcontrolador ATMEGA8 no podría conectarse al “módulo Bluetooth” por tener una Interfaz Serial USART. La verdad es que USART es un dispositivo de comunicación serial flexible y de alto nivel, en otras palabras, USART es completamente compatible con AVR UART. (Atmel Corporation)
- 23 Puertos Programables para entradas y salidas I/O; ya que el proyecto está planificado para manejar alrededor de 14 pines entre entradas y salidas. Los

23 pines del ATMEGA8 son más que suficientes para controlar los dispositivos del vehículo.

- 4.5V – 5.5V, son los voltajes de polarización del microcontrolador, este rango hay que tomar en cuenta al momento de planificar el sistema de alimentación para el circuito eléctrico de control.
- Consumo eléctrico en estado Activo: 3.6mA, en Modo Idle: 1.0mA; estas características de consumo son importantes al momento de implementar el circuito electrónico en el vehículo, ya que su bajo consumo de operación, no permite que la batería del vehículo llegue a descargarse.
- Pin “Reset”, este puerto funciona polarizándolo con 5V más una resistencia de 1K a 10K ohm. El pin se activa a través de un pulsador o contactor a tierra.

2.8 Compilador Bascom y simulador Proteus

Al igual que Visual Studio 2008, Bascom-AVR y Proteus VSM, forman parte de las herramientas esenciales para el desarrollo de las diferentes aplicaciones que se implementarán en el circuito electrónico. La placa a implementarse en el vehículo Mini Austin contiene varios dispositivos electrónicos que se describirán en la siguiente sección.

2.8.1 Bascom-AVR

BASCOM-AVR, es un compilador potente y fácil de usar para la serie AVR de microcontroladores desarrollados por Atmel (ver figura 16). El programa contiene una interfaz de usuario amigable y un set de comandos simples, convirtiéndolo así, en un programa más flexible de usar que otros compiladores. Cualquier persona con un conocimiento básico de C o C++ puede escribir un programa con

éxito utilizando Bascom, ya que la mayoría de sus funciones y estructura de declaración, son similares a los de C. Adicionalmente, en el internet existen varios tutoriales gratuitos [1], que ayudan a aprender y a programar microcontroladores. Para los programadores profesionales, ellos encontrarán muchos comandos muy conocidos y enlaces/bucles que acortarán las líneas de programación, reduciendo así, la memoria flash o EEPROM requerida.

El programa simulador, el comprobador de sintaxis y las herramientas de emulación pueden ayudar a explorar el programa en busca de errores antes de grabarlo o quemarlo en cualquier microcontrolador. Para grabar sobre microcontroladores, se necesita un ordenador o una laptop con puerto USB; se compila y se guarda el programa como un archivo *.HEX o *.BIN, y finalmente, se graba o se quema utilizando quemadores de flash USBISP. (Informer Technologies párr. 1-3)

- [1] http://avrhelp.mcselec.com/
- http://www.mcselec.com/index2.php?option=com_forum&Itemid=59

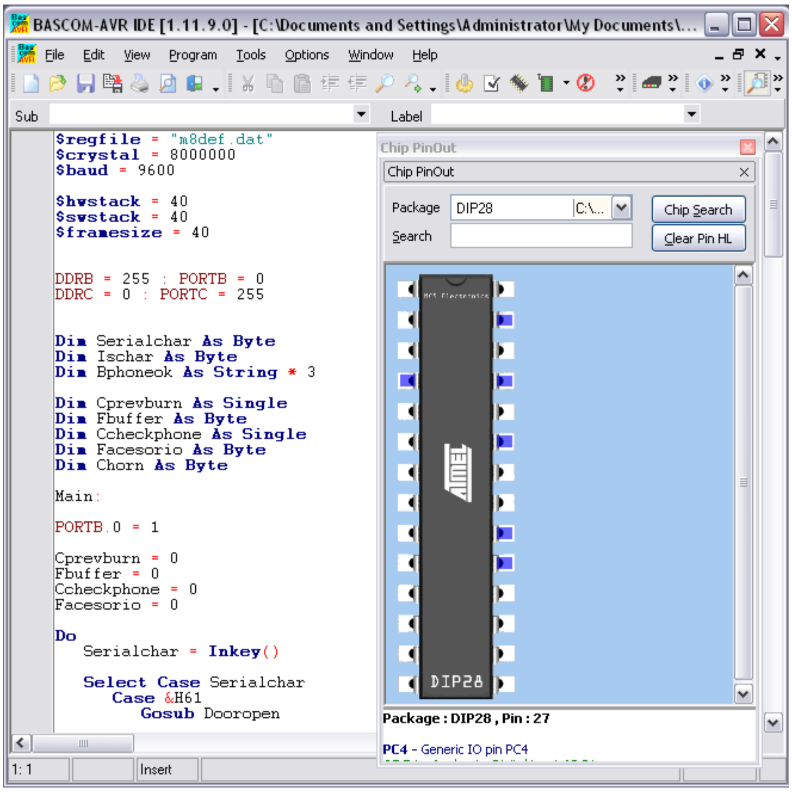


Figura 16. Software BASCOM-AVR.

2.8.2 Proteus VSM

El programa Proteus VSM (Virtual System Modelling), es una herramienta para la verificación vía software que permite comprobar, prácticamente en cualquier diseño, la eficacia del programa desarrollado por el usuario (ver Figura 17). Su combinación de, simulación de código de programación, y de su simulación mixta SPICE, permite verificaciones analógico-digitales de sistemas basados en microcontroladores de una manera eficiente.

El entorno de diseño electrónico PROTEUS VSM de Labcenter Electronics ofrece la posibilidad de simular código microcontrolador de alto y bajo nivel, en modo mixto de SPICE. Los tres entornos que permiten realizar estas simulaciones son: ISIS para el diseño gráfico, VSM para la simulación y el ARES para el diseño de

placas. (García Breijo, Compilador C CSS y Simulador PROTEUS para Microcontroladores PIC VII: 168)

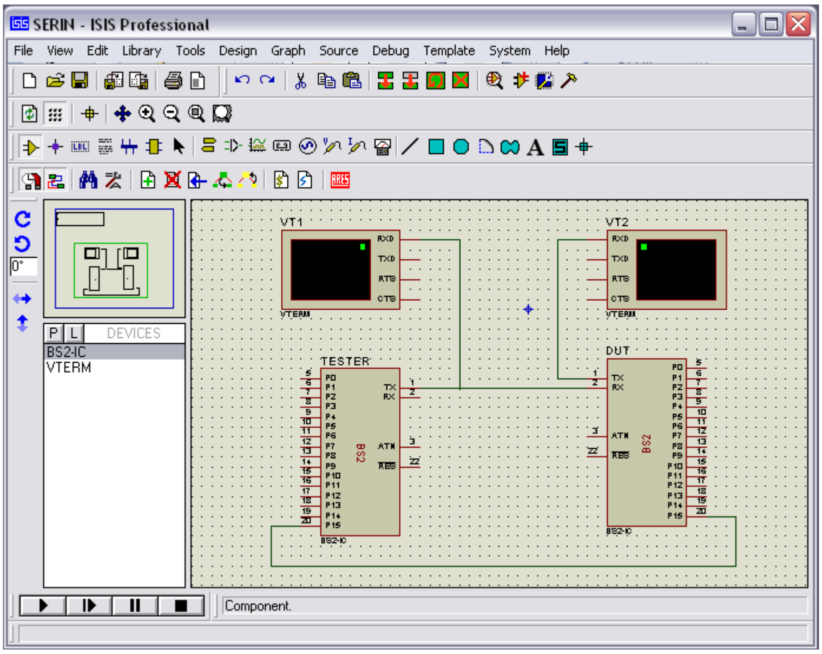
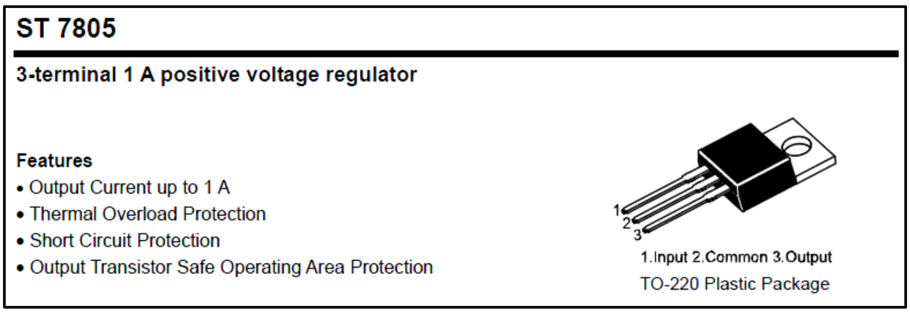


Figura 17. Software Proteus VSM.

2.9 Circuito electrónico de control

Los principales elementos electrónicos y electromagnéticos que forman parte del circuito electrónico de control son: reguladores de voltaje, transistores bipolares y relés electromagnéticos.

2.9.1 Circuito Integrado, regulador de voltaje



ST 7805

3-terminal 1 A positive voltage regulator

Features
- Output Current up to 1 A
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Figura 18. Características de un Regulador de Voltaje. (Semtech Electronics Ltd. 1)

Los Circuitos Integrados (C.I.) son dispositivos semiconductores que agrupan una serie de transistores, diodos, resistores, capacitores, etc. dentro de un mismo chip de silicio (ver Figura 18). Los C.I. más empleados en las placas de control son los reguladores de tensión, amplificadores, temporizadores, control de procesos y funcionamiento de un equipo (microcontroladores). El “Regulador de Tensión” así como el “Diodo Zener”, se usan como regulador; sin embargo, para el desarrollo de esta tesis se utilizaron los C.I. por ser más precisos y dar protección contra corto circuitos (Cisilino 73).

El esquema de conexión típico de un regulador de tensión es el mostrado a continuación (ver Figura 19).

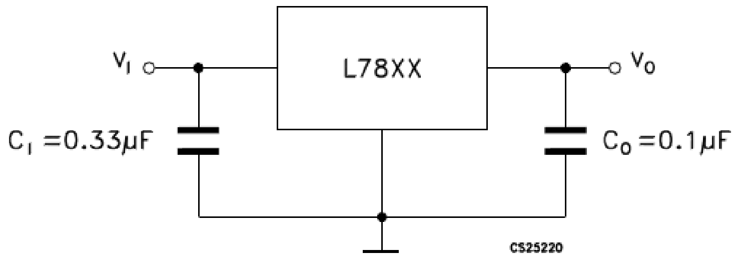


Figura 19. Circuito típico de aplicación de un Regulador de Voltaje. (Cisilino 73)

Este circuito es muy fácil de construir y es común para toda la línea de los reguladores 78xx. La Tabla 4, se describe los principales 78xx clasificados en función a la tensión que pueden entregar (Cisilino 74).

Regulador	Tensión de Salida	Tensión de Entrada Mínima	Máxima
μA7805C	5	7	25
μA7806C	6	8	25
μA7808C	8	10,5	25
μA7810C	10	12,5	28
μA7812C	12	14,5	30
μA7815C	15	17,5	30
μA7818C	18	21	33
μA7824C	24	27	38

Tabla 4. Reguladores de tensión positiva. (Tensión expresadas en Voltios). (Cisilino 74)

2.9.2 Transistores bipolares

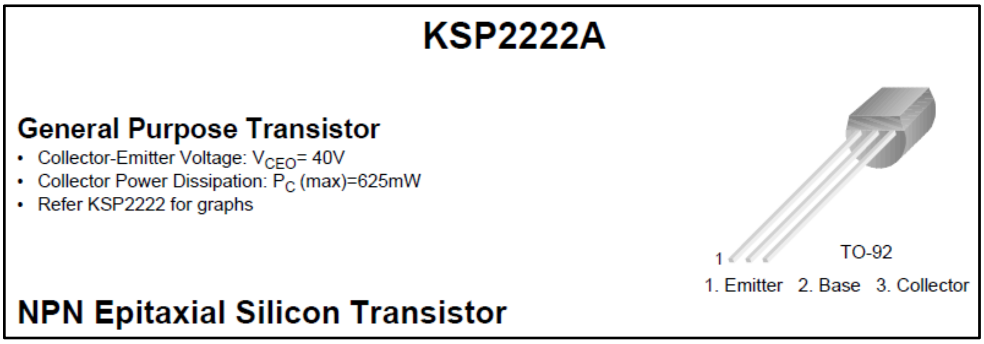


Figura 20. Características Transistor N-P-N. (Semtech Electronics Ltd. 1)

Los transistores son dispositivos semiconductores de estado sólido, generalmente fabricados con silicio, al que se agregan impurezas. Los transistores tienen distintas denominaciones en base a su tipo de construcción, pero el común denominador es que tienen tres terminales (ver figura 20). Los transistores bipolares tienen la propiedad de variar la corriente que circula a través de ellos mediante una polarización muy pequeña, es decir, se pueden manejar grandes corrientes mediante la inyección de una corriente de saturación apropiada. Este es el principio por el cual los transistores son muy utilizados como elementos amplificadores de potencia.

El Transistor Bipolar es uno de los transistores más utilizados. Consta de tres bloques de material semiconductor, que se pueden disponer en configuración NPN o PNP, y de tres terminales, Base, Colector y Emisor.

Las denominaciones NPN y PNP corresponden al tipo de material con el que están dopados los bloques de silicio. Estos bloques en realidad son uno solo, el secreto es que al agregarle impurezas en lugares precisos, se producen zonas dentro del bloque, delimitadas por junturas. Esto permite que tenga propiedades semiconductoras. (Rodríguez 32)

Para el proyecto de tesis se aprovechó el funcionamiento del transistor como una especie de válvula. Permite controlar el flujo de corriente cuando ésta intenta pasar a través de él (ver Figura 21).

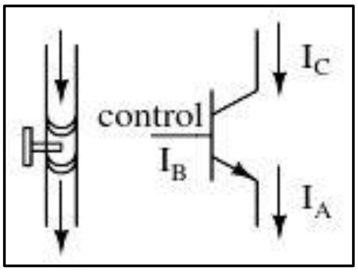


Figura 21. Modelo simple de transistor N-P-N como una válvula. (Arrayás 281)

Para entender mejor al transistor como una válvula se resumirá a continuación el funcionamiento de los transistores N-P-N mediante cuatro reglas (para los P-N-P simplemente hay que cambiar las polaridades):

- El colector debe ser más positivo que el emisor, esto es $V_C > V_E$.
- Si se consideran la unión base-emisor y la unión base-colector como dos diodos, (ver Figura 22), entonces el diodo base-emisor debe estar polarizado

directamente y el diodo base-colector debe estarlo inversamente. Esto es, $V_B = V_E + 0,6$ y $V_C > V_B$.

- Un transistor se estropea si se superan sus valores máximos de I_C, I_B, V_{CE} y V_{BE}.
- Cuando las reglas anteriores se cumplen, entonces $I_C = \beta I_B$. (Arrayás 281)

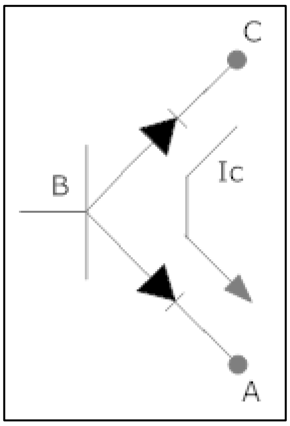


Figura 22. Manera en cómo un ohmímetro ve un transistor N-P-N. (Arrayás 282)

2.9.3 Relés electromagnéticos

Es un dispositivo que se utiliza para conectar dos circuitos independientes por los que circulan corrientes de voltajes diferentes. El relé está formado por un interruptor y un electroimán, y cuando es accionado por la corriente de un circuito de entrada, éste actúa de interruptor del otro circuito. (Larousse Editorial, S.L.)

Con la llave S_1 abierta (ver Figura 23), la tensión no excita a la bobina; por lo tanto, el relé no se acciona y los contactos quedan en la posición de reposo, impidiendo la tensión de alimentación de la lámpara, la cual no encenderá.

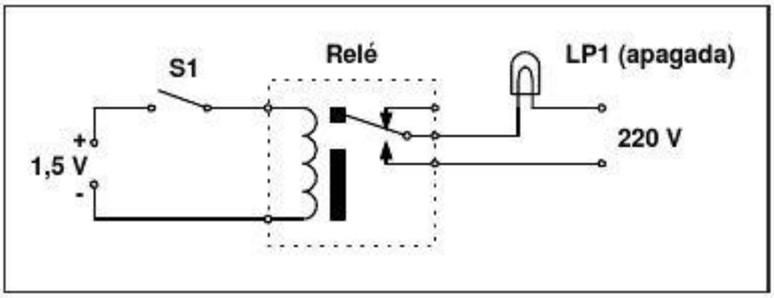


Figura 23. Relé no excitado. (Cisilino 60)

Si se excita la bobina con voltaje directo (ver Figura 24), se provoca el cierre del contacto eléctrico, y la lámpara se encenderá (Cisilino 60).

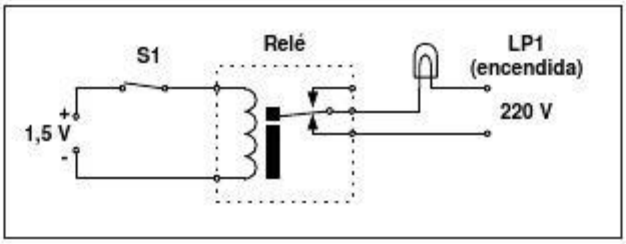


Figura 24. Relé excitado. (Cisilino 61)

Los reguladores de voltaje, los transistores bipolares, los relés electromagnéticos y otros componentes eléctricos, conforman los elementos del circuito electrónico a implementarse en el automóvil, siendo éste a su vez, la interface hardware con el cual el dueño del vehículo controlará el sistema de encendido y los servomotores de las puertas del mismo.

CAPÍTULO 3

3. Metodología

3.1 Programación en Visual Studio

Para describir mejor el programa realizado en Visual Studio, se presenta a continuación los diagramas de flujo del mismo.

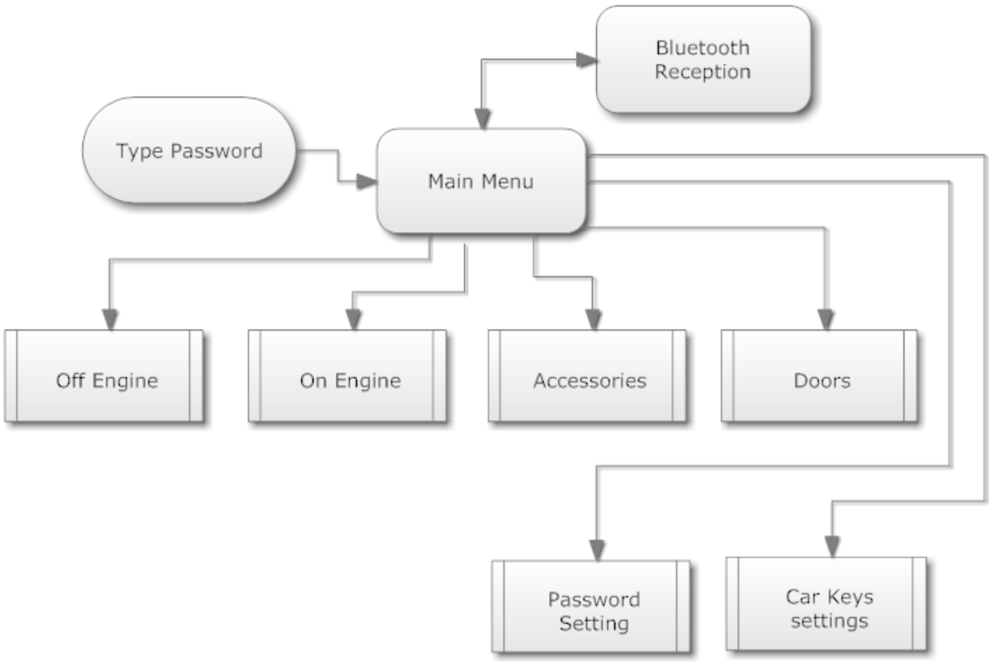


Figura 25. Diagrama de flujo Visual Studio, subpanel “Main Menu”.

Se ha implementado un submenú de seguridad en donde se debe introducir una clave alfanumérica, cada vez que el celular se conecta al módulo Bluetooth del vehículo.

Adicionalmente, un proceso alterno de Recepción Bluetooth corre todo el tiempo, permitiendo que el programa reciba datos sin colgarse. Para evitar estos errores se implementa programación por hilos (ver figura 23).

Después de ingresar correctamente la clave del programa y conseguir la respectiva conexión con el módulo Bluetooth, se ingresa al Menú Principal el cual contiene las funciones que el sistema de alarma puede ejecutar.

3.1.1 Subpanel "Off Engine"

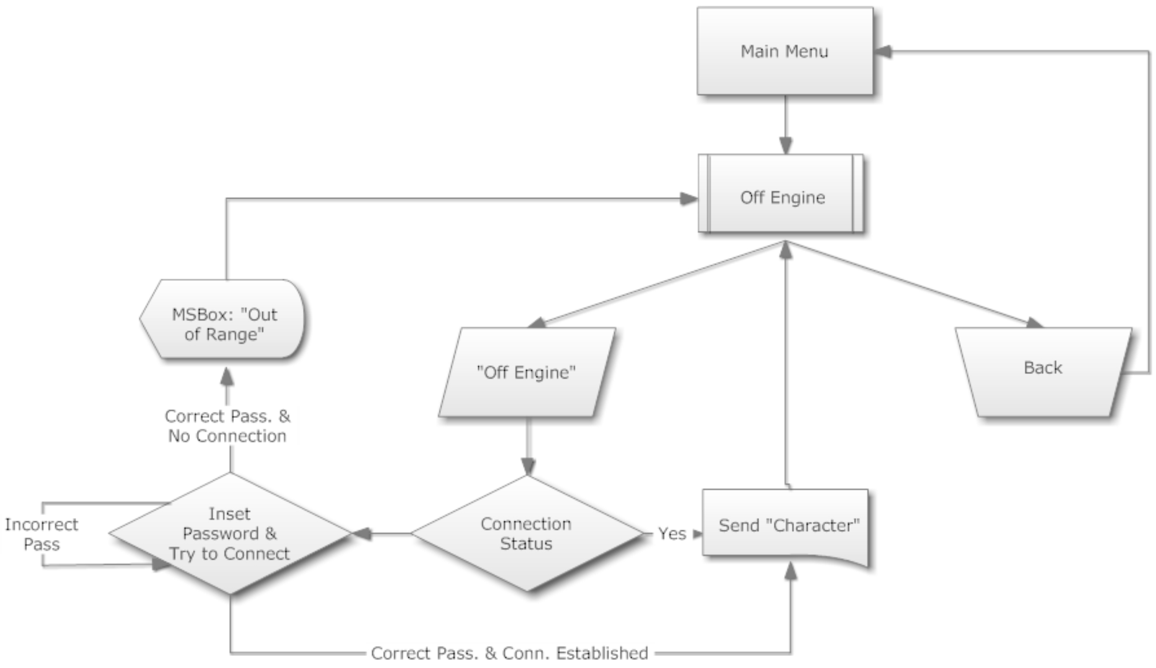


Figura 26. Diagrama de flujo, Subpanel "Off Engine".

En el subpanel "*Off Engine*" se tiene dos opciones, hacer click en "Back", el cual retorna a Menú Principal o hacer click en "Off Engine", el cual enviará la indicación de "apagar el vehículo" al módulo Bluetooth siempre y cuando exista una conexión pre-establecida. En el caso que no la haya, el programa entra en un proceso de verificación de clave y conexión Bluetooth. Dependiendo de los resultados exitosos o no, continua o descarta la tarea de enviar la indicación respectiva (ver Figura 24).

3.1.2 Subpanel "On Engine"

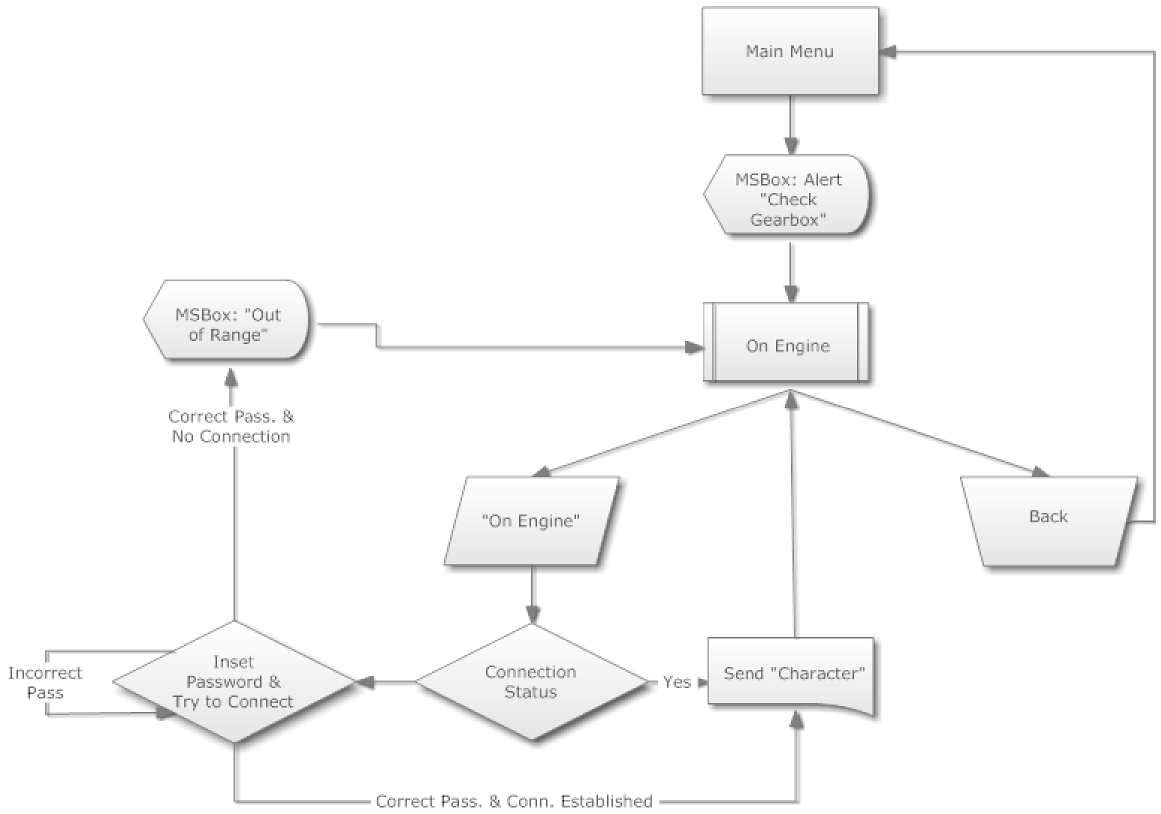


Figura 27. Diagrama de flujo, subpanel "On Engine".

Antes de ingresar al sub-panel "On Engine", se levanta una Alerta "Revisar que la caja de cambios se encuentre en Neutro". Después de confirmar el mensaje, en el subpanel "On Engine" se tiene dos opciones, hacer click en "Back", el cual retorna a Menú Principal o **pulsar** en "On Engine". Al aplastar el botón "On Engine", se envía la orden de iniciar el arranque del motor del auto, y cuando se suelta el botón, se envía la orden de suspender el arranque al módulo Bluetooth siempre y cuando exista una conexión pre-establecida. En el caso que no la haya, el programa entra en un proceso de verificación de clave y conexión

Bluetooth; dependiendo de los resultados exitosos o no, continua o descarta la tarea de enviar las indicaciones respectivas (ver Figura 27).

3.1.3 Subpanel "Accessories"

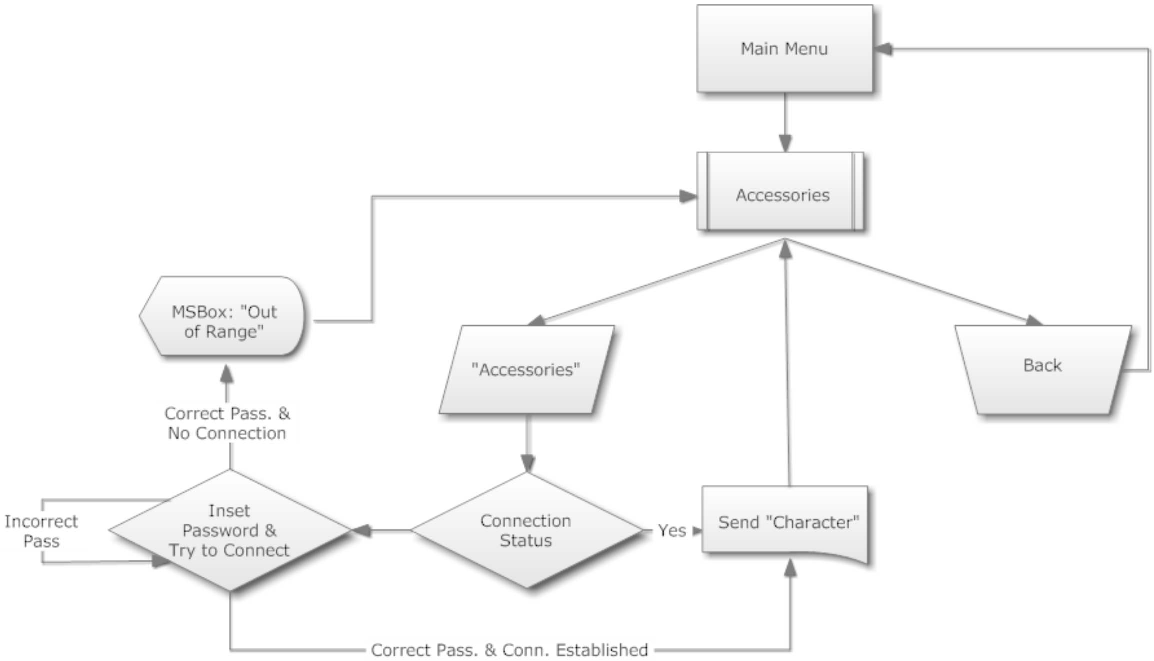


Figura 28. Diagrama de flujo, subpanel "Accessories".

En el subpanel "Accessories" se tiene dos opciones, hacer click en "Back", el cual retorna a Menú Principal o hacer click en "Accessories", el cual enviará la indicación de "activar o desactivar la función de Accesorios" al módulo Bluetooth siempre y cuando exista una conexión pre-establecida. En el caso que no la haya, el programa entra en un proceso de verificación de clave y conexión Bluetooth; dependiendo de los resultados exitosos o no, continua o descarta la tarea de enviar la indicación respectiva (ver Figura 28).

3.1.4 Subpanel “Doors”

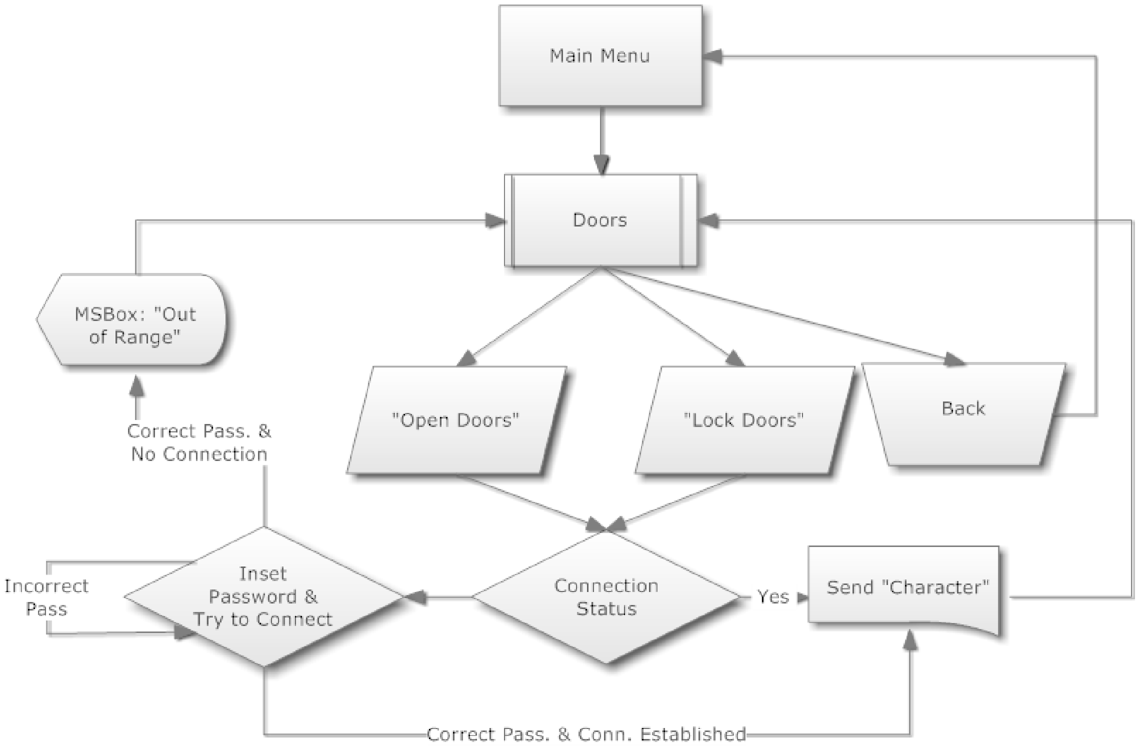


Figura 29. Diagrama de flujo, subpanel “Doors”.

En el subpanel “Doors” se tiene tres opciones, hacer click en “Back”, el cual retorna a Menú Principal o hacer click en “Open Doors” o en “Lock Doors”, el cual enviará la indicación de “activar los servomotores de las puertas” al módulo Bluetooth siempre y cuando exista una conexión pre-establecida. En el caso que no la haya, el programa entra en un proceso de verificación de clave y conexión Bluetooth; dependiendo de los resultados exitosos o no, continua o descarta la tarea de enviar la indicación respectiva (ver Figura 29).

3.1.5 Subpanel "Password Setting"

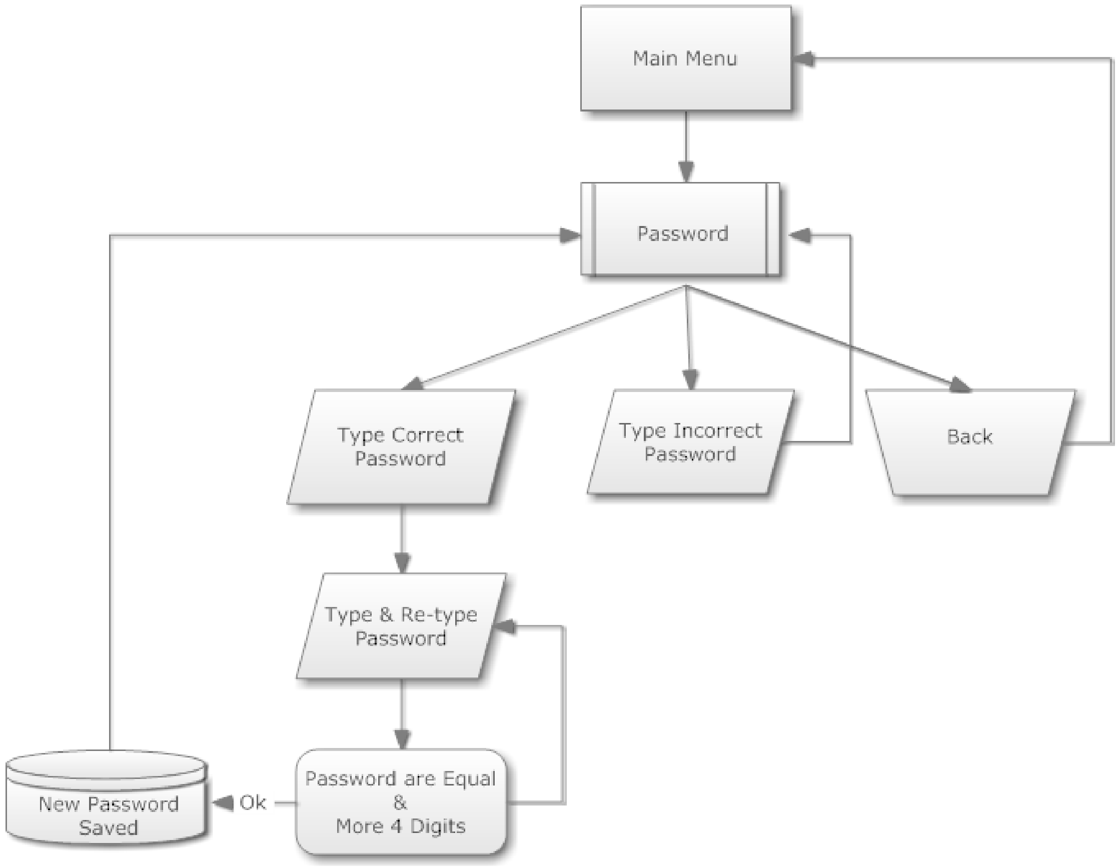


Figura 30. Diagrama de flujo, subpanel "Password Setting".

El subpanel "Password Setting", tiene el botón "Back", el cual retorna al "Menú Principal". Adicionalmente, existe la casilla para ingresar la clave actual con el objetivo de autentificar al dueño, quien es la persona que quiere cambiar la clave. Posteriormente al ingreso correcto de la clave, aparecen dos casillas para comprobar que la nueva clave ingresada está correctamente escrita y sea mayor a cuatro dígitos. Cumpliendo estas dos premisas, se guarda de manera encriptada la nueva clave y se retorna al menú "Password Setting", caso contrario solicita que reingrese la nueva clave correctamente (ver Figura 30).

3.1.6 Subpanel “Car Keys Settings”

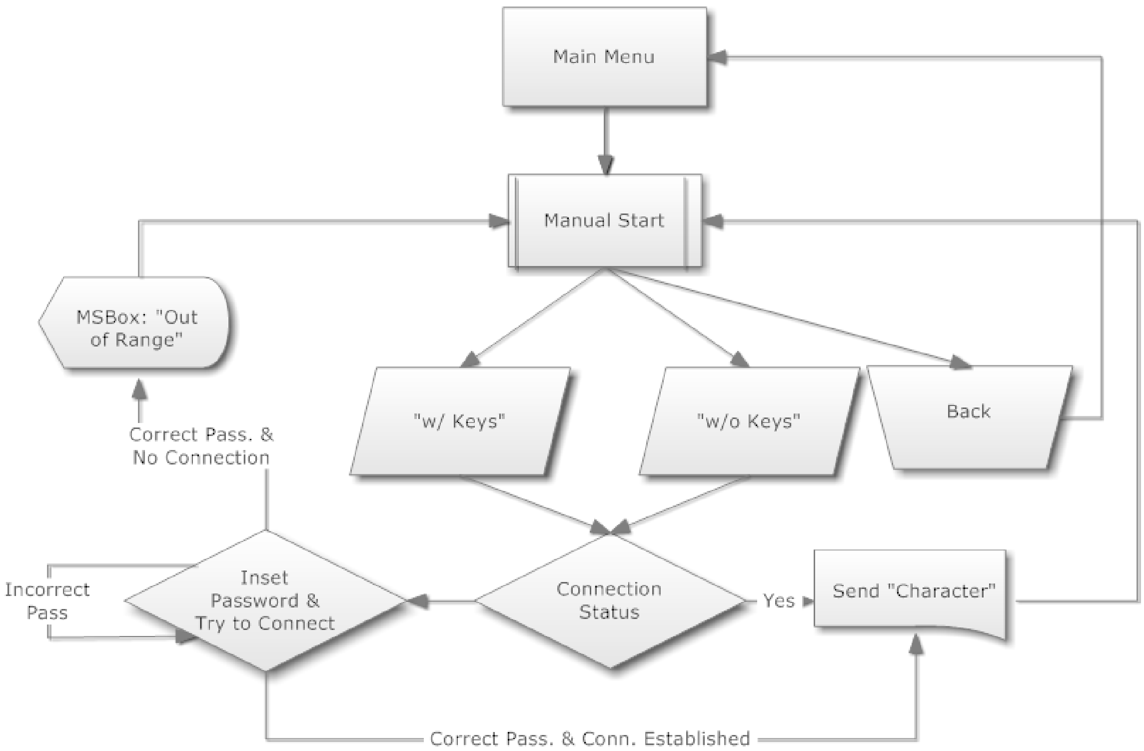


Figura 31. Diagrama de flujo, subpanel “Car Keys Settings”.

En el subpanel “Car Keys Settings” se tiene tres opciones, hacer click en “Back”, el cual retorna a Menú Principal o hacer click en “w/ Keys” o en “w/o Keys”, el cual enviará la indicación de “activar o desactivar el cilindro de arranque” al módulo Bluetooth, siempre y cuando exista una conexión pre-establecida. En el caso que no la haya, el programa entra en un proceso de verificación de clave y conexión Bluetooth; dependiendo de los resultados exitosos o no, continua o descarta la tarea de enviar la indicación respectiva (ver Figura 31).

3.2 Propiedades y configuración del puerto serial

En esta sección se revisará como se conecta la aplicación programada con el dispositivo Bluetooth del *smartphone* y las propiedades del puerto serial de Visual Basic.

Existe un elemento en común que permite que el Bluetooth del teléfono y la aplicación programada en Visual Basic puedan conectarse, es el puerto de comunicación. Lo que se intenta es configurar el mismo puerto tanto para el dispositivo Bluetooth del *smartphone,* como para el programa de Visual Basic.

A continuación se indicarán los pasos a seguir para conectar el dispositivo Bluetooth con el puerto de comunicación 2 (COM2). En la pantalla “Bluetooth Devices”, ubicarse sobre “FireFly-1559” y hacer click en “Menu” y dirigirse a “COM Ports” (ver Figura 32).



Figura 32. Pantalla “Bluetooth Devices”, opción “COM Ports”.

A continuación, aparece “adicionar un nuevo puerto de comunicación”, se hace click en “Menu” (ver Figura 33).

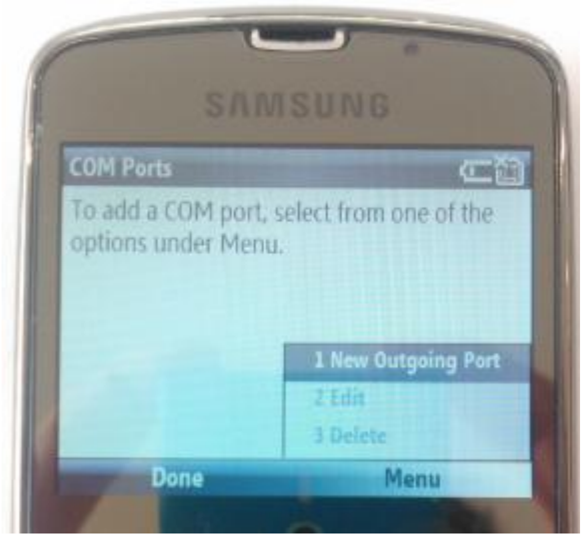


Figura 33. Pantalla “COM Ports”, opción “New Outgoing Port”.

Posteriormente se selecciona “New Outgoing Port”, se escoge COM2, se deja chequeado la opción “Secure Connection” y se hace click en “Done” (ver Figura 34).



Figura 34. Pantalla “COM Ports”, opción COM2.

Una vez finalizada la configuración del puerto de comunicación para el dispositivo Bluetooth del teléfono inteligente, se realiza la misma operación en la aplicación de Visual Basic. En la parte de diseño, se utiliza el recurso “SerialPort”, y en sus propiedades, se modifica el *PortName* por COM2 (ver Figura 35). No es estricto que el puerto 2 siempre se utilice, lo importante es que sea el mismo puerto tanto en la configuración del puerto del celular, como en la aplicación Visual Basic.

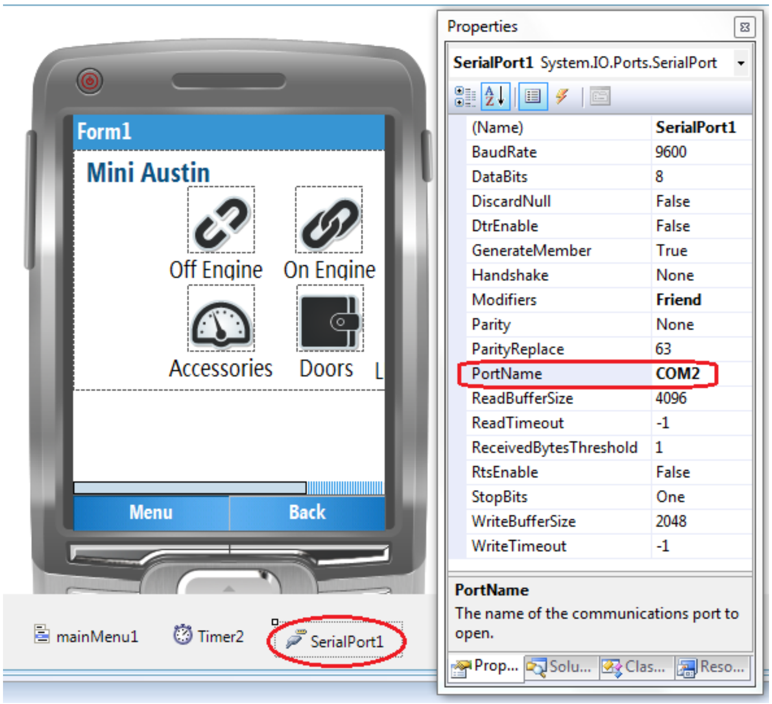


Figura 35. Configuración de puerto de comunicación en Visual Studio.

Otro tema importante para lograr conexiones exitosas entre el *smartphone* y el módulo Bluetooth RN-41, es que las propiedades del puerto serial de ambos dispositivos sean iguales. A continuación se muestran las propiedades más importantes para la aplicación Visual Basic (ver Figura 36).

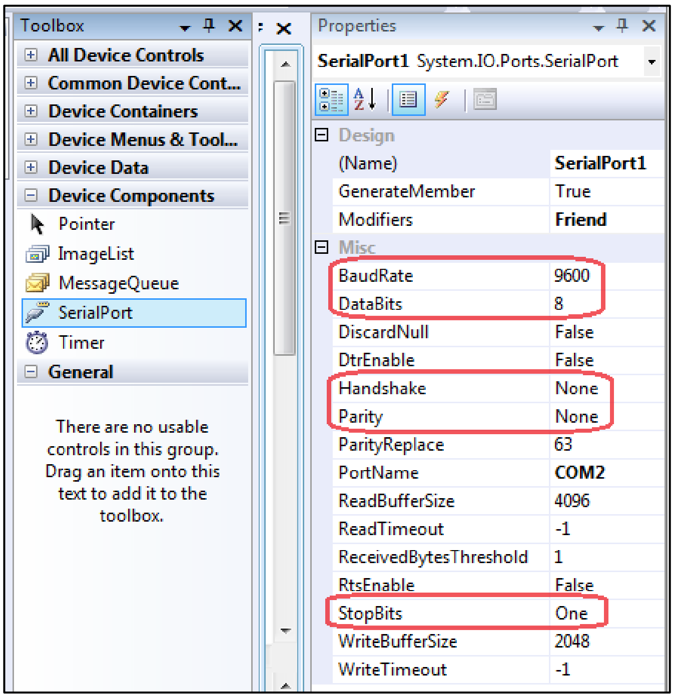


Figura 36. Propiedades del puerto serial en Visual Studio.

3.3 Proceso para ejecutar la aplicación Interface Gráfica de Usuario en el *smartphone*

Una vez realizada en Visual Studio de la programación de la Interface Gráfica de Usuario, la configuración del puerto de comunicación y en el *smartphone* la configuración del puerto serial mencionados en la sección anterior, se procede a cargar el programa en el teléfono inteligente. Los prerrequisitos para realizar la conexión del celular con el IDE Visual Basic, es tener precargado el software "Sync Center" y poseer el cable de conexión USB del equipo móvil hacia la computadora.

Cuando se conecta el celular a la computadora con el cable USB, automáticamente se sincroniza el celular con Visual Basic; sin embargo, en caso de no hacerlo existe un botón que permite la sincronización de manera manual.

Una vez sincronizado el celular, se hace click en “Start Debugging” (ver Figura 37), después, se escoge la opción que no es el emulador (ver Figura 38), y finalmente la aplicación se carga en el celular (ver Figura 39.)

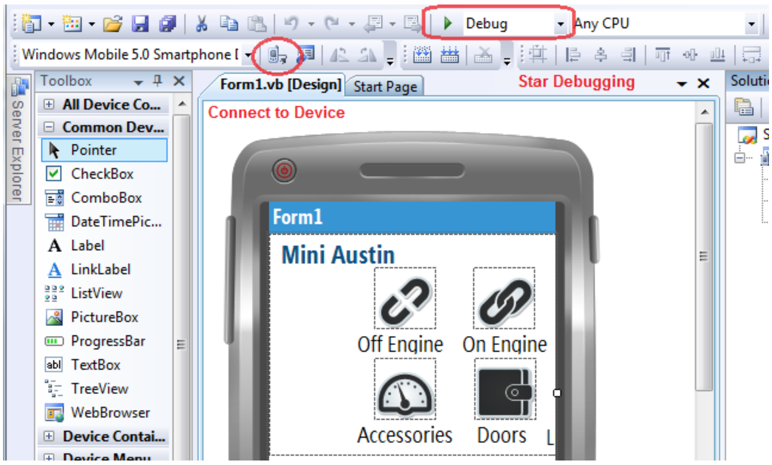


Figura 37. Proceso para ejecutar la aplicación en el *smartphone*, Paso 1.

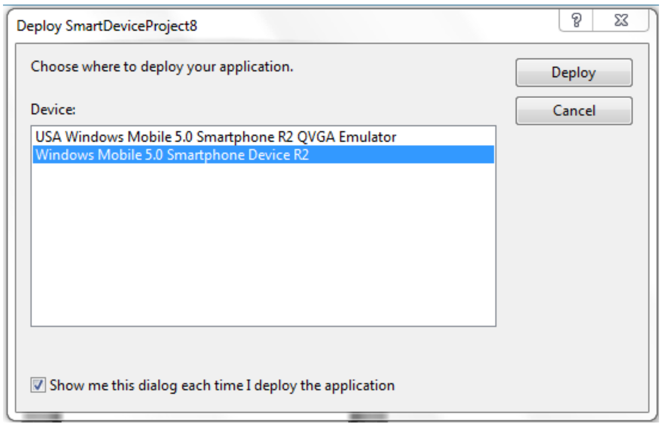


Figura 38. Proceso para ejecutar la aplicación en el *smartphone*, Paso 2.

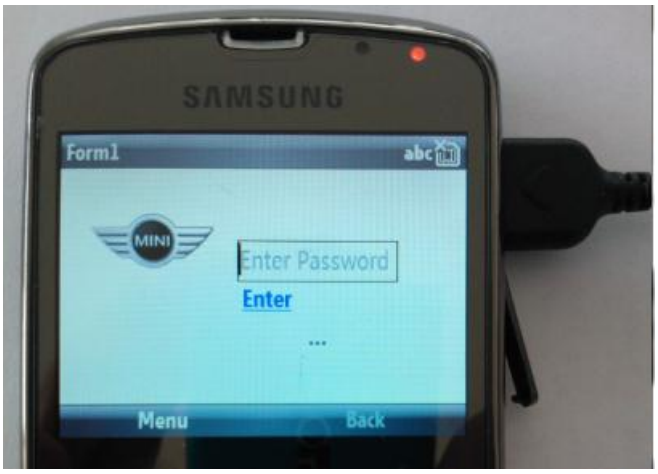


Figura 39. Aplicación ejecutándose sobre el *smartphone.*

Para volver a correr la aplicación en el celular sin recurrir a conectar al computador, se ejecuta el archivo de la aplicación “Tesis” desde la carpeta (\Program Files\Tesis_Andres). La carpeta con el nombre “Tesis_Andres” puede ser reemplazada por el nombre que el usuario desee (ver Figura 40).



Figura 40. Carpeta donde se encuentra el archivo ejecutable de la aplicación.

3.4 Programación módulo Bluetooth RN-41

La programación del Módulo Bluetooth (xBT) se hace a través del cable DB-9 de una PC, el cual se conecta a un circuito Integrado MAX-232 para que convierta las señales RS-232 a pulsos TTL, y finalmente se conecta al módulo RN-41 o Bluetooth (Roving Networks , Advanced 3), como lo indica el diagrama a continuación (ver Figura 41).

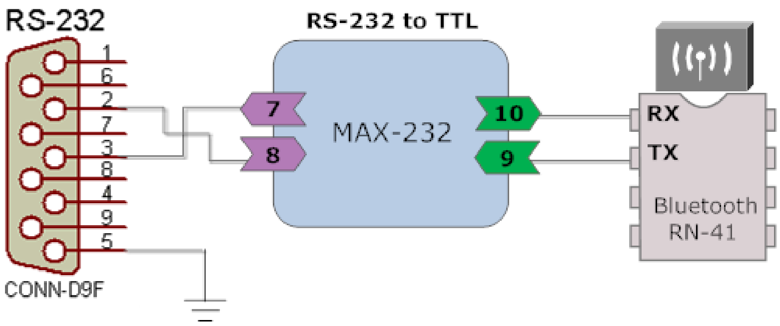


Figura 41. Conexión física para programar módulo Bluetooth.

Una vez concluida la conexión física, se configura el emulador HyperTerminal como se indica a continuación (ver Figura 42).

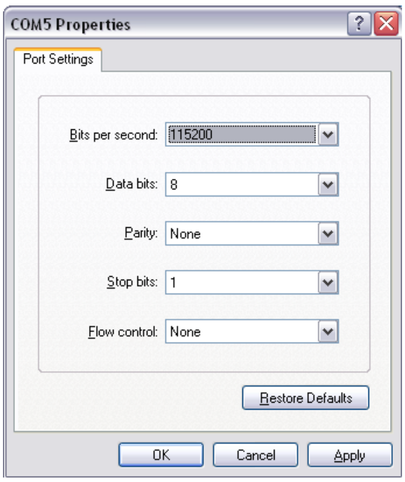


Figura 42. Propiedades HyperTerminal para conectarse módulo Bluetooth.

Hecha la conexión física y la comunicación en el HyperTerminal, se escribe en la pantalla "**$$$**" e instantáneamente debe aparecer "**CMD**", esto verifica que el cable y la configuración de comunicación están correctos (Roving Networks , Advanced 5), (ver Figura 43).

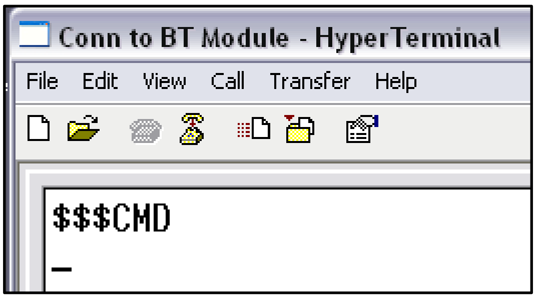


Figura 43. HyperTerminal y confirmación de conexión exitosa con módulo Bluetooth.

Los parámetros de configuración que se implementaron en el Módulo Bluetooth se resumen en la tabla a continuación (ver Tabla 5).

Comando	Categoría	Descripción
$$$	Ingresar	Ingresa al Command mode del Módulo xBT
SU,96	Configuración	Configuración BaudRate a 9600
SY,0001	Configuración	Potencia de salida a 6 dBM, ≈ 15m de cobertura
SR,Z	Autenticación	Borra el registro de las direcciones xBT
I	Autenticación	Busca dispositivos cercanos y despliega sus direcciones xBT
SR,I	Autenticación	Guarda la última dirección xBT desplegada
SX,1	Seguridad extra	*Bonding*, solo acepta conexiones de direcciones xBT guardadas anteriormente. 1 activa, 0 desactiva
SE,1	Seguridad	*Encryption*, 1 activa, 0 desactiva
SP,123456	Seguridad	Security *Pin code*
SI,0000	Seguridad extra	*Invisible*, desactiva periodo para solicitar ingreso a Módulo xBT
D	Información	Despliega Configuración Básica
E	Información	Despliega Configuración Extendida
R,1	Reiniciar	Reinicia el Módulo xBT

Tabla 5. Parámetros de configuración de módulo Bluetooth.

3.5 Programación ATMEGA8

Para describir mejor el programa realizado en Bascom, se presenta a continuación el diagrama de flujo del mismo.

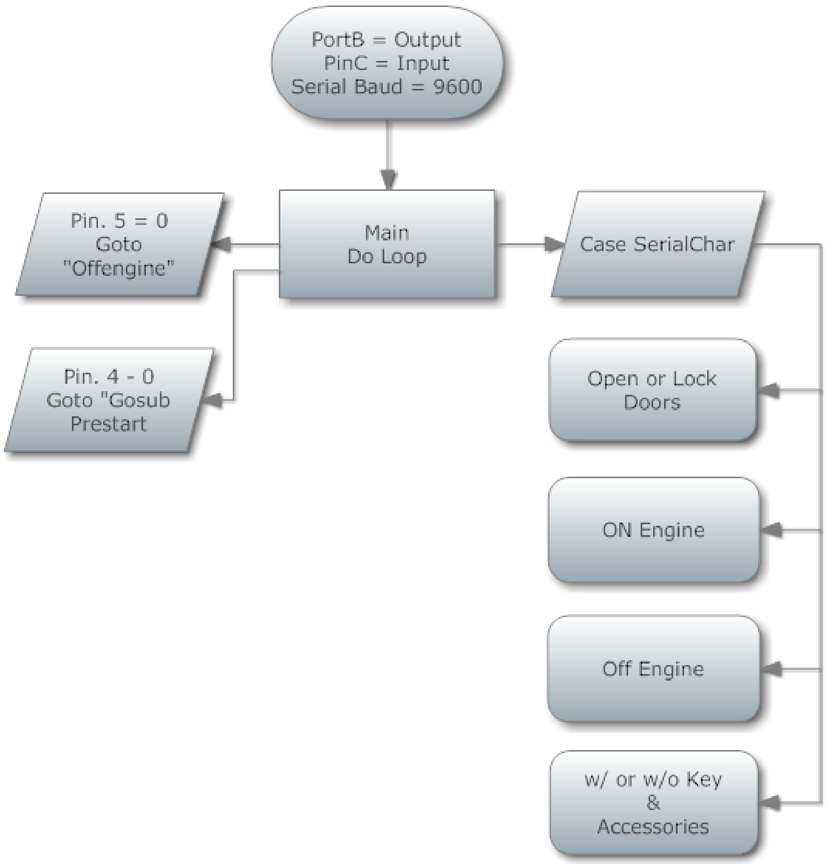


Figura 44. Diagrama de flujo microcontrolador, “Main Menu”.

El programa inicia con la configuración de los puertos de entrada-salida y los pines de comunicación serial. Luego pasa al “Menú Principal”, éste es un bucle “Do Loop”, el cual está constantemente evaluando las alternativas “Select Case” o “IF End”. Cuando una de estas instrucciones es verdadera, el microcontrolador pasa a ejecutar la instrucción contenida en dicha condición. Cabe indicar que las alternativas “Select Case” o ”IF End” se activan ya sean por interrupciones en

sus pines de entrada, o datos recibidos por su puerto serial. Los pines de entrada activan las subrutinas “OffEngine” u “OnEngine (PreStart)”. El puerto serial dependiendo del dato recibido, activa las subrutinas de “Open or Close Door”, “OnEngine”, “OffEngine”, ”Accessories”, “activación o desactivación de llaves”. A continuación se presentan las funciones o acciones que ejecutarán las distintas subrutinas del microcontrolador.

3.5.1 Subrutinas “Off Engine”

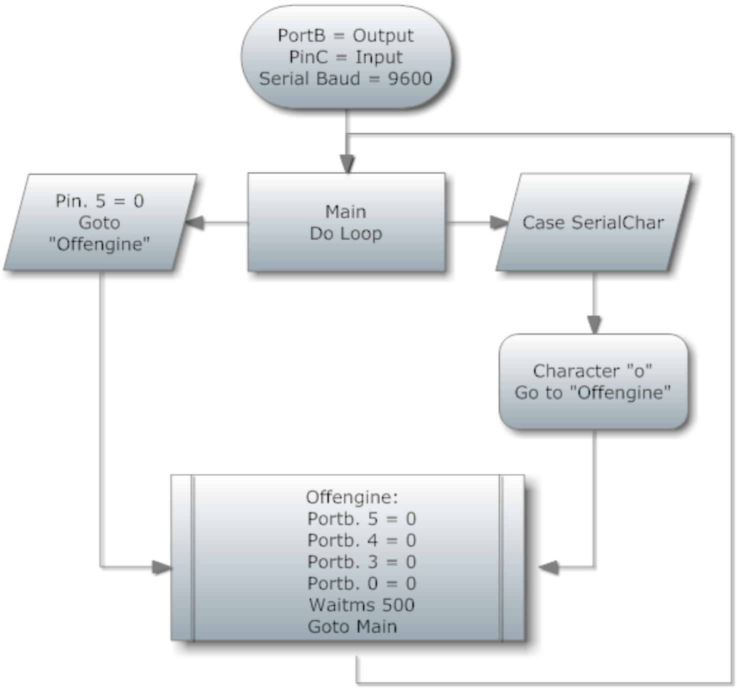


Figura 45. Subrutinas “Off Engine”.

El microcontrolador activa la subrutina “OffEngine” cuando identifica ya sea el PIN.5 en bajo o, el carácter “o” en su puerto serial. La acción que ejecuta es apagar sus puertos, cambiando a circuito abierto los transistores, que cuando estaban en circuito cerrado, activaban a los relés que permitían que el vehículo esté prendido.

3.5.2 Subrutina “On Engine”

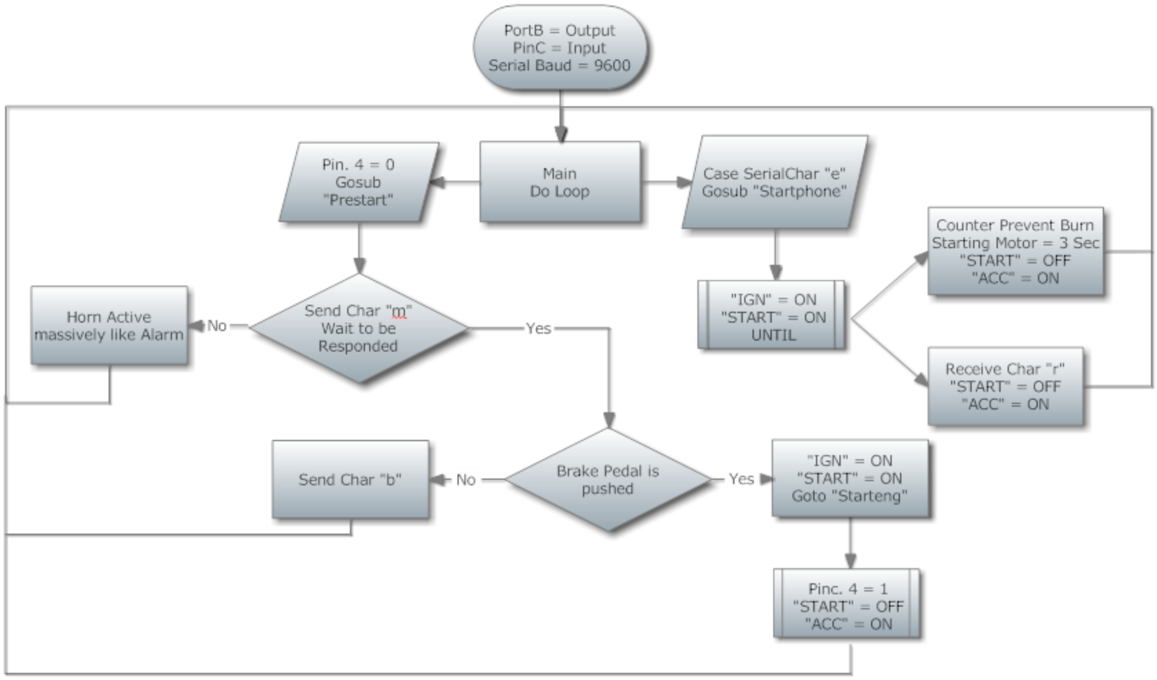


Figura 46. Subrutinas “On Engine”.

El motor del vehículo puede encenderse de dos maneras: remotamente desde el celular, o a través del botón “START” instalado en el tablero del mismo.

Desde el celular, se envía la letra “e” al microcontrolador, y éste activa la señal al motor de arranque para iniciar el encendido del motor. Para detener el motor de arranque existe dos formas: la primera es recibiendo la letra “r”, la cual apaga la acción “START”, y la segunda es en el momento que un contador interno llega a 3 segundos y desactiva automáticamente el motor de arranque. Esta opción de los tres segundos, evita que el motor de arranque dañe al motor por no recibir la letra “r” desde el módulo Bluetooth o del celular.

Cuando se presiona el Botón “Start”, antes de activar el motor de arranque, se acciona un sistema de seguridad antirrobo para comprobar que el dueño se encuentra dentro del vehículo. Para llevar a cabo dicha evidencia, después de presionar el botón “Start”, el microcontrolador, envía una letra al celular, el cual contesta automáticamente. En caso que la respuesta sea negativa, el automóvil activa la bocina varias veces a modo de alarma. En el caso que sea positiva, el proceso que verifica si el freno se encuentra presionado inicia. Si no lo está, el microcontrolador envía un mensaje al celular indicando que debe aplastarse el freno para poder encender el auto. Si el freno está accionado, el motor de arranque enciende el motor del vehículo hasta que el usuario deje de aplastar el botón “Start” (ver Figura 46).

Las condiciones de seguridad como validar la presencia del *smartphone* y comprobar que el pedal de freno esté presionado, se basan en los siguientes motivos:

- El vehículo no enciende si el dueño no está portando el celular consigo. Esto se hace con el objetivo de evitar que un delincuente intente encender el automóvil y robárselo.
- En los vehículos modernos, si no está presionado el pedal del embrague, el auto no enciende; sin embargo, en este proyecto, el arranque del Mini Austin no puede producirse con el embrague presionado, por lo que se simuló la acción de aplastar un pedal utilizando el freno.

3.5.3 Subrutinas “Open & Lock Doors”, “w/ or w/o keys” y “Accessories”

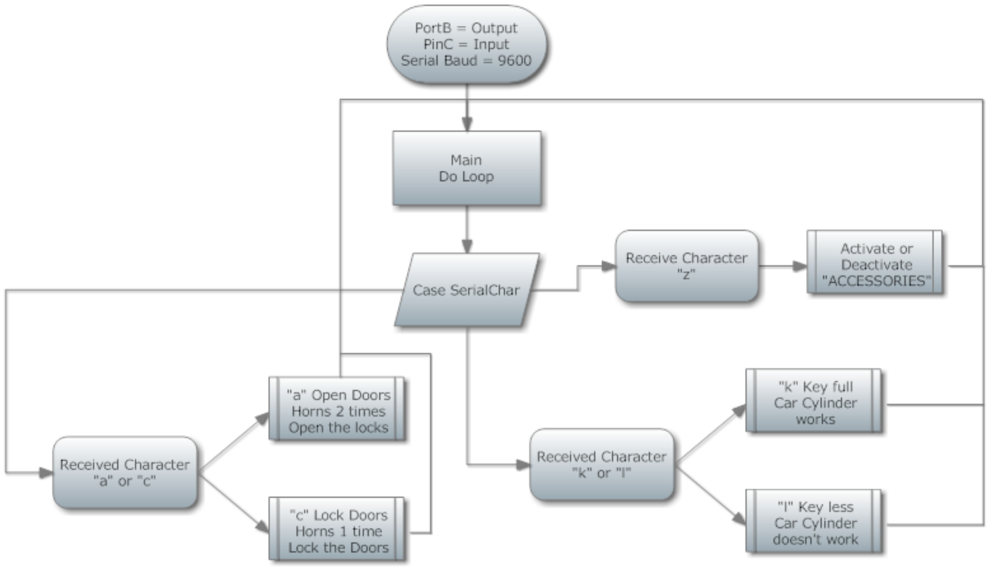


Figura 47. Subrutinas “Open & Lock Doors”, “w/ or w/o keys” y “Accessories”.

La subrutina “Open & Lock Doors” se planificó como una alarma estándar. Cuando el usuario desea abrir los seguros de las puertas, el microcontrolador activa la bocina dos veces y después activa el servomotor para cerrar los seguros. Cuando se desea cerrar los seguros de las puertas, la bocina se activa una vez y después se activa el servomotor para cerrar los seguros.

En esta misma subrutina se deseó implementar la acción de cerrar la ventana cada vez que se activa los seguros de las puertas; sin embargo, instalar el mecanismo de vidrios eléctricos en la puerta del Mini Austin no fue posible por motivos de espacio.

La subrutina de “Accessories” se implementa ante la necesidad de encender los dispositivos eléctricos como la radio, las luces, etc., sin tener que encender el vehículo.

La subrutina de “w/ or w/ Keys” activa y desactiva el funcionamiento del “cilindro de arranque” del vehículo a través de un arreglo de relés implementados en el circuito electrónico de control. Esta subrutina surge ante la necesidad de dejar el vehículo en el taller para su mantenimiento, pero sin tener que dejar el celular del dueño. Esta función permite que el automóvil arranque el motor utilizando el sistema tradicional de las llaves.

3.6 Simulaciones en Proteus

Para verificar que las distintas subrutinas programadas en el microcontrolador (µC) ATMEGA8 cumplan con los objetivos trazados, se realizaron simulaciones en el Simulador ISIS de Proteus (ver Figura 48). Los resultados de dichas simulaciones se resumen en la siguiente tabla y en los videos a continuación. (ver Tabla 6 y Videos 1 al 8).

Los videos están grabados en el CD anexo al documento impreso de la tesis.

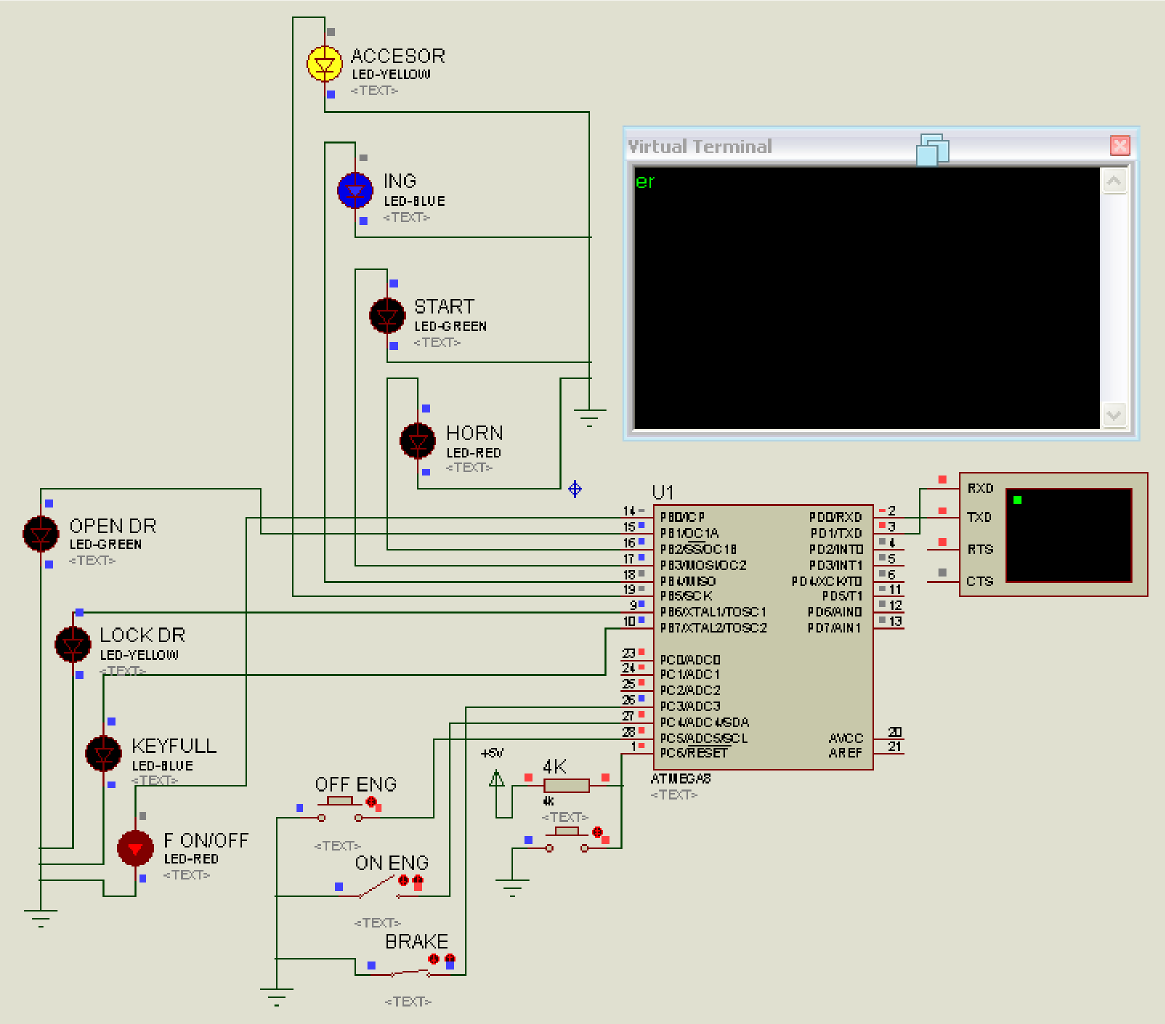


Figura 48. Simulaciones del µC sobre ISIS de Proteus.

ATMEGA8				
Puertos de Entrada				
Interrupción	**Subrutina**	**Descripción**	**Acción**	**Video**
PIN.5 = 0	OffEngine	Apagar vehículo	Apaga todos los **LEDs**, excepto KeyFull	Video 1
PIN.4 = 0	OnEngine	Revisa Conn. Smarphone	Envía **"M"** y espera recibir **"n".** Si no recibe **"n"**, **LED Horn** se activa 3 veces	Video 2
	PreStart1	Revisa Freno activado	"StartEngine" o Envía **"B"**	Video 3
	StarEngine	Encender el vehículo	LEDs se activan, **ING-START**, **ING-ACC**	
Puerto Serial				
Dato Recibido	**Subrutina**	**Descripción**	**Acción**	
"o"	OffEngine	Apagar vehículo	Apaga todos los **LEDs**, excepto KeyFull	Video 4
"e"	ONEngine	Encender el vehículo	Enciende **LEDs ING-START**, y pasa a **ING-ACC** cuando desborda tiempo de precaución o recibe **"r"**	Video 5
"a" o "c"	"Open" o "Close" Door	Abrir o Cerrar las Puertas	Encender **LEDs**, **Horn** y **Open DR** y **Lock DR**	Video 6
"k" o "l"	w/ o w/o Keys	Activa Función de llaves	Activa o desactiva **LED KeyFull**	Video 7
"z"	Accessories	Activa Accesorios	Activa o desactiva **LED ACCESSORIES**	Video 8

Tabla 6. Resultados de la simulación de subrutinas en microcontrolador.

A continuación se presentan los videos con las subrutinas activadas a través de los pines del puerto de entrada:

http://youtu.be/dnkGo211mio

Video 1. Subrutina "OffEngine" activada por el pin de entrada del µC.

http://youtu.be/qPulqG35B5s

Video 2. Subrutina "OnEngine" activada por el pin de entrada del µC.

http://youtu.be/7Vtr215MUKs

Video 3. Subrutina "Prestar y StartEngine" activada por el pin de entrada del µC.

A continuación los videos con las subrutinas activadas a través del puerto serial:

http://youtu.be/tid2Pgf3DNw

Video 4. Subrutina “Offengine” activada a través del puerto serial del µC.

http://youtu.be/P-Eo3CzYWCo

Video 5. Subrutina “OnEngine” activada a través del puerto serial del µC.

http://youtu.be/SQyYTwT6qFE

Video 6. Subrutina “Open-Close Door” activada a través del puerto serial del µC. (2)

http://youtu.be/5Y3nko0tf_s

Video 7. Subrutina “w/ or w/o keys” activada a través del puerto serial del µC.

http://youtu.be/LNMCJOI_nYk

Video 8. Subrutina “Accessories” activada a través del puerto serial del µC.

[2] El software que se utilizó para grabar la pantalla del computador, no alcanzó a registrar la manera en cómo los diodos se encendían y se apagaban rápidamente.

3.7 Grabación del microcontrolador con "PROGISP"

Finalizada la etapa de pruebas sobre el microcontrolador en Proteus, se pasa a la etapa final de grabación con el programa de "PROGISP".

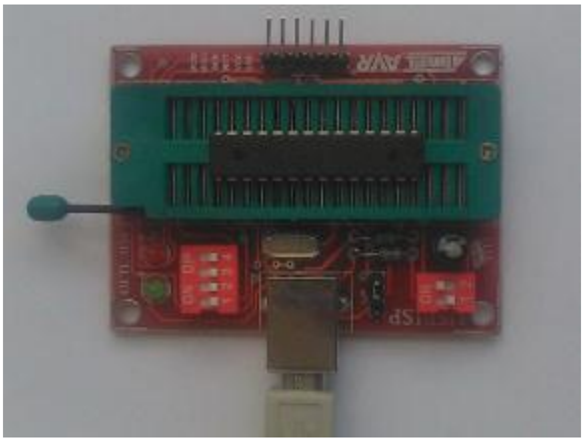


Figura 49. Microcontrolador y circuito de grabación "PROGISP".

Para grabar el programa se coloca el microcontrolador sobre el socket del circuito de grabación y se conecta el cable USB al computador (ver Figura 49). Posteriormente en la interface de usuario se siguen los siguientes pasos: primero, se escoge el tipo de microcontrolador, en este caso es el ATmega8; segundo, haciendo click en "Load Flash", se abre una ventana donde se escoge el archivo *.HEX del proyecto a ser grabado en el microcontrolador, y tercero se hace click en "Auto", donde se graba automáticamente el archivo *.HEX escogido (ver figura 50).

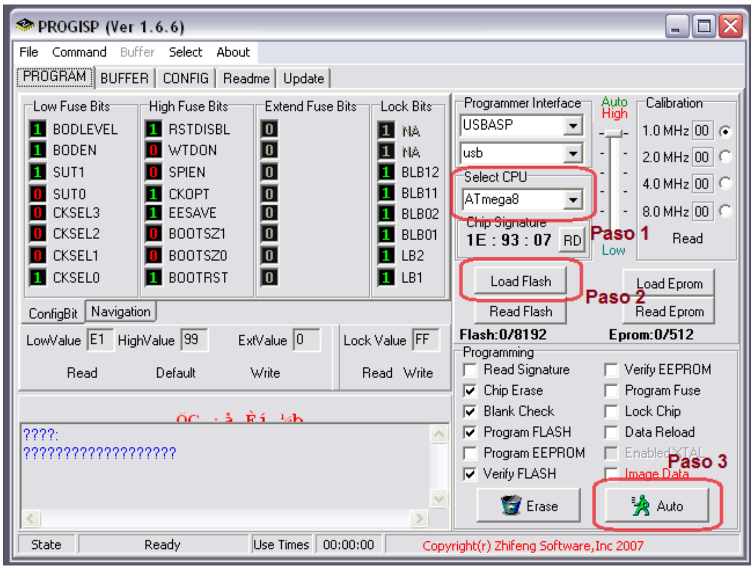


Figura 50. Grabación microcontrolador con “PROGISP”.

3.8 Diseño del circuito impreso

El circuito electrónico es la interface entre los componentes del automóvil y los puertos de entrada y salida del microcontrolador. Para comprender mejor el esquema y los dispositivos que lo componen, se ha realizado el diagrama a continuación (ver Figura 51).

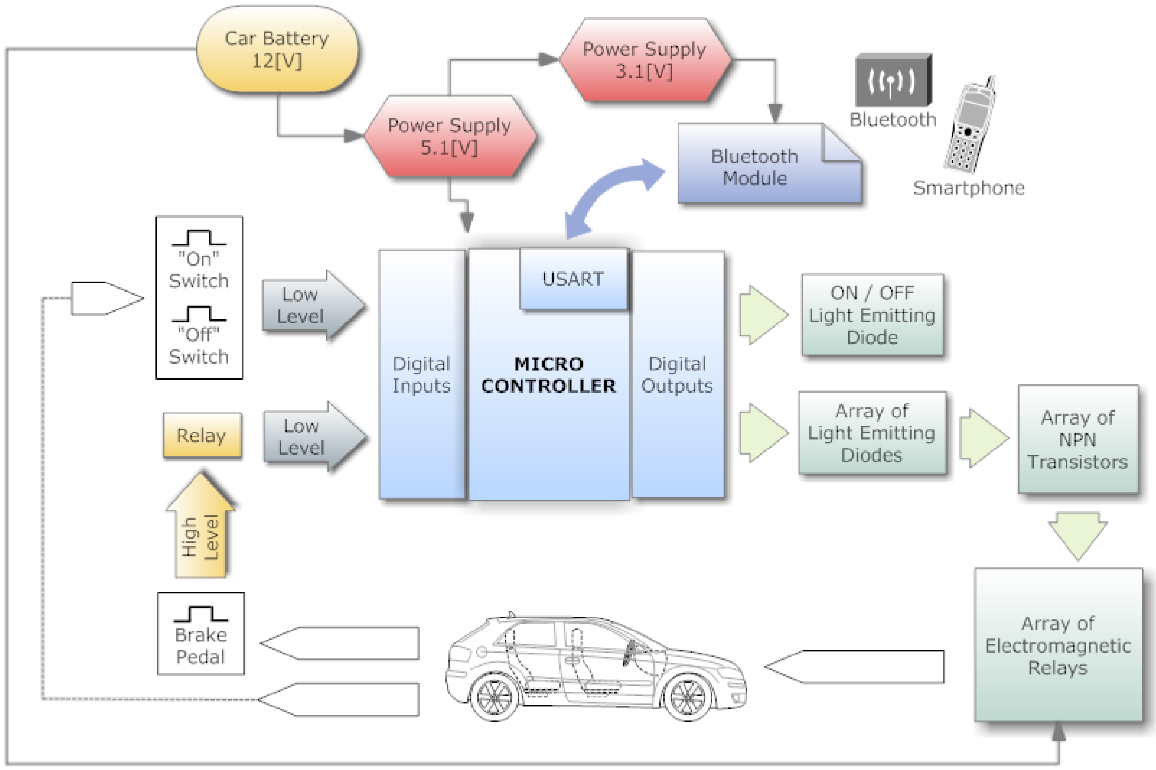


Figura 51. Diagrama del esquema del circuito electrónico.

El circuito electrónico se divide en los siguientes segmentos:

- Divisores de voltaje y dispositivos de seguridad eléctricos.
- Módulo Bluetooth.
- Microcontrolador, puertos entrada/salida.
- Arreglo de transistores NPN y diodos emisores de luz.
- Arreglo de relés electromagnéticos.

3.8.1 Divisores de voltaje y dispositivos de seguridad electrónicos

El circuito tiene que activar o alimentar tres segmentos: el microcontrolador, el módulo Bluetooth y al arreglo de los relés electromagnéticos, este último funciona con 12V, tensión que es tomada de la batería del automóvil.

El circuito electrónico posee dos reguladores de voltaje, el “7805 SC” y el “L117-3.3”. El primero regulador entrega 5V y una corriente de máxima de 1A, el cual sirve para alimentar al microcontrolador. El otro regulador alimenta al módulo Bluetooth con 3.2V. Adicionalmente, para mejorar la protección en caso de sobre voltajes o tensiones inversas, se instalaron en paralelo a tierra dos diodos Zener, uno protege al microcontrolador y el otro protege al módulo Bluetooth. (ver Figura 56).

3.8.2 Circuito electrónico, módulo Bluetooth

El módulo Bluetooth es instalado en el circuito de control con el objetivo de establecer una conexión entre el microcontrolador y el *smartphone* a través de sus pines y antena RF. Los pines uno y dos levantan la comunicación serial UART con el microcontrolador, mientras que la antena establece la comunicación RF con el *smartphone* (ver Figura 56).

3.8.3 Circuito electrónico, microcontrolador, puertos de entrada y salida

Los puertos de entrada se activan en “nivel bajo”, esto significa que, el microcontrolador identifica un “pin activado” cuando éste está conectado a tierra. Para lograr que los pines “ON” y “OFF” alcancen tierra, éstos se conectaron directamente a la masa del vehículo. Para que el pin “freno” alcance tierra, éste necesita una conexión adicional de un relé pequeño.

Cuando el trompo del freno es presionado, éste entrega 12V que son aprovechados por el relé para mover sus contactores a tierra; de esta manera, el

microcontrolador identifica el pin “freno” se encuentra activado (ver Figura 51, 56 y 57).

Los pines de salida se utilizan para activar los transistores y los diodos LED. El voltaje del pin en “nivel alto” es 4.2V a 2.2V, y en “nivel bajo” el voltaje es 0V.

3.8.4 Arreglo de transistores NPN y diodos emisores de luz

Para seleccionar el transistor adecuado se deben considerar varios *inputs*, uno de ellos, es saber cuál es la corriente y el voltaje del relé con el que se va a trabajar. El relé seleccionado para este proyecto, se explica a detalle en la sección 3.8. El resto de *inputs* son las corrientes y voltajes de saturación o activación del transistor (I_B y R_B), los cuales se detallan a continuación.

Por el concepto $I_C = \beta \times I_B$, I_C = Corriente del Relé, e I_B = Corriente del pin del microcontrolador: La corriente nominal del relé seleccionado es 110mA y la corriente que el microcontrolador puede entregar es de 40mA (Atmel Corporation 235). Entonces el factor de ganancia β es el siguiente:

$$\beta = \frac{Ic}{I_B} = \frac{0.11 \text{ A}}{0.04 \text{ A}} = 2.75$$

Ecuación 1. Factor de ganancia del transistor.

A continuación, se busca en el mercado un transistor que tenga un β igual a 2.75 y una potencia de disipación de 720mW [3]. El transistor que se puede encontrar

---

[3] La potencia de consumo del relé es 360mW. Por motivos de seguridad y eficiencia se escoge un transistor con una potencia de disipación mayor a 360mW. Para este proyecto, se decidió duplicar esta protección a 720mW.

en el mercado con las condiciones indicadas es el KSP2222A, el cual tiene un factor de ganancia β = 35 y una potencia de disipación de 625mW. Presentadas estas nuevas condiciones, se calcula la nueva corriente I_B que se necesita para saturar el transistor.

$$\beta = \frac{Ic}{I_B} \;\; => \;\; I_B = \frac{I_C}{\beta} = \frac{0.11\,A}{35} = 0.003143A = 3.1mA$$

Ecuación 2. Nueva corriente base I_B para saturar el transistor.

Esta corriente cumple con la restricción de no sobrepasar la corriente máxima que el microcontrolador puede entregar, 3.1mA < 40mA. A continuación se calcula la resistencia de la base R_B para obtener la corriente de saturación I_B (ver Figura 52).

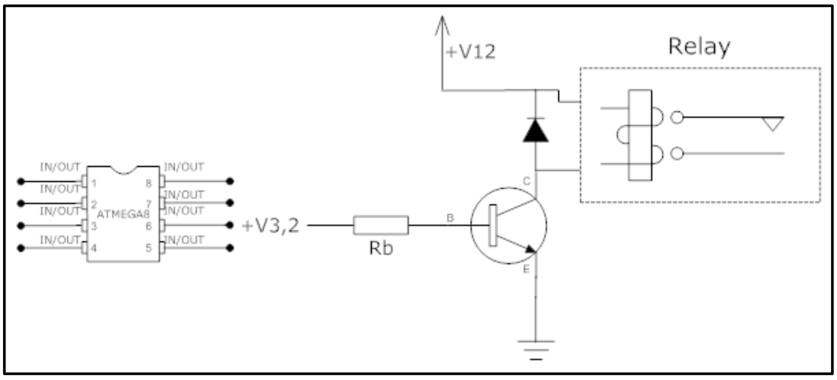


Figura 52. Diagrama Transistor, calculo Rb.

$$R_B = \frac{V}{I_B} = \frac{(3.2 - 0.6)V}{3.1mA} = 838.7\Omega$$

Ecuación 3. Cálculo de R_B.

El voltaje 3.2V pertenece al voltaje promedio que el pin del microcontrolador entrega según el *datasheet*, que es entre 4.2V a 2.2V. El voltaje 0.6 pertenece al voltaje de saturación Base-Emisor del transistor.

En función de los Inputs escogidos y calculados, P_C = 625mW, I_B = 3.1mA y R_B = 840Ω, se realizan simulaciones para comprobar que el transistor KSP2222A es el adecuado (ver Figura 53).

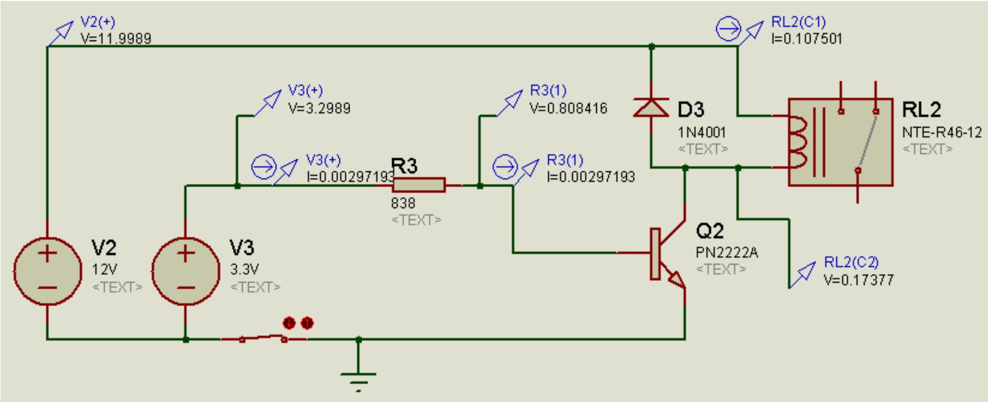


Figura 53. Simulación sobre el transistor N2222A.

La simulación indica que el transistor no se quemará ya que no se sobrepasa ninguna de sus capacidades máximas del transistor, I_C = 625mA, I_B = 15mA, V_{CE} = 0.3V y V_{BE} = 1.2V (Arrayás 281).

Para mejorar la presentación del circuito e identificar el instante que un pin del puerto de salida del microcontrolador se activa o se desactiva, se implementaron diodos emisores de luz o LED. Para incluir a cada pin un LED, se separa la resistencia de 838Ω en dos resistencias en serie para hacer una división de voltaje. De esta manera se entrega suficiente corriente al transistor y suficiente voltaje al diodo foto-lumínico (ver Figura 54).

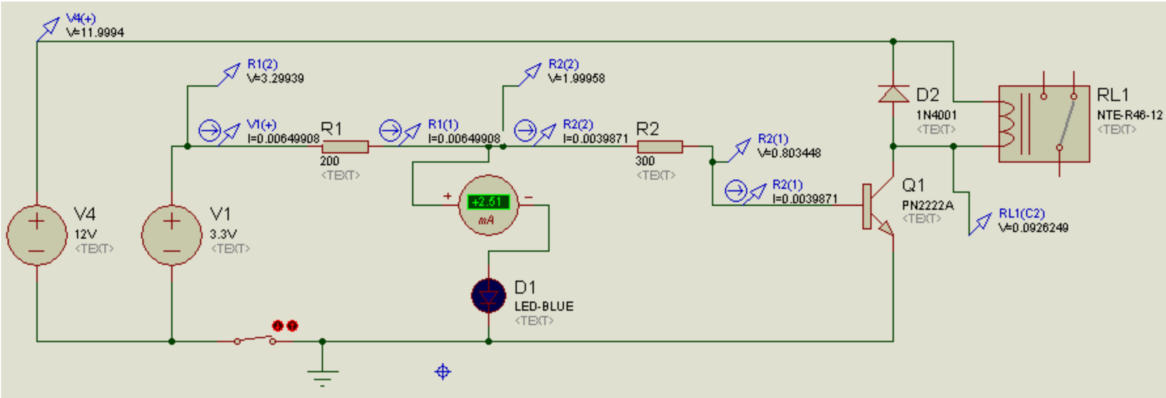


Figura 54. Conexión transistor N2222A con LED y relé.

3.8.5 Arreglo de relés electromagnéticos

Con la situación de conectar circuitos de alto amperaje y bajo amperaje, los relés a implementarse son la interface de conexión entre el sistema eléctrico del automóvil de alto amperaje y el circuito electrónico diseñado de bajo amperaje (ver Figura 47).

Para explicar mejor las aplicaciones que los relés van a controlar, se ha hecho el siguiente diagrama (ver Figura 55).

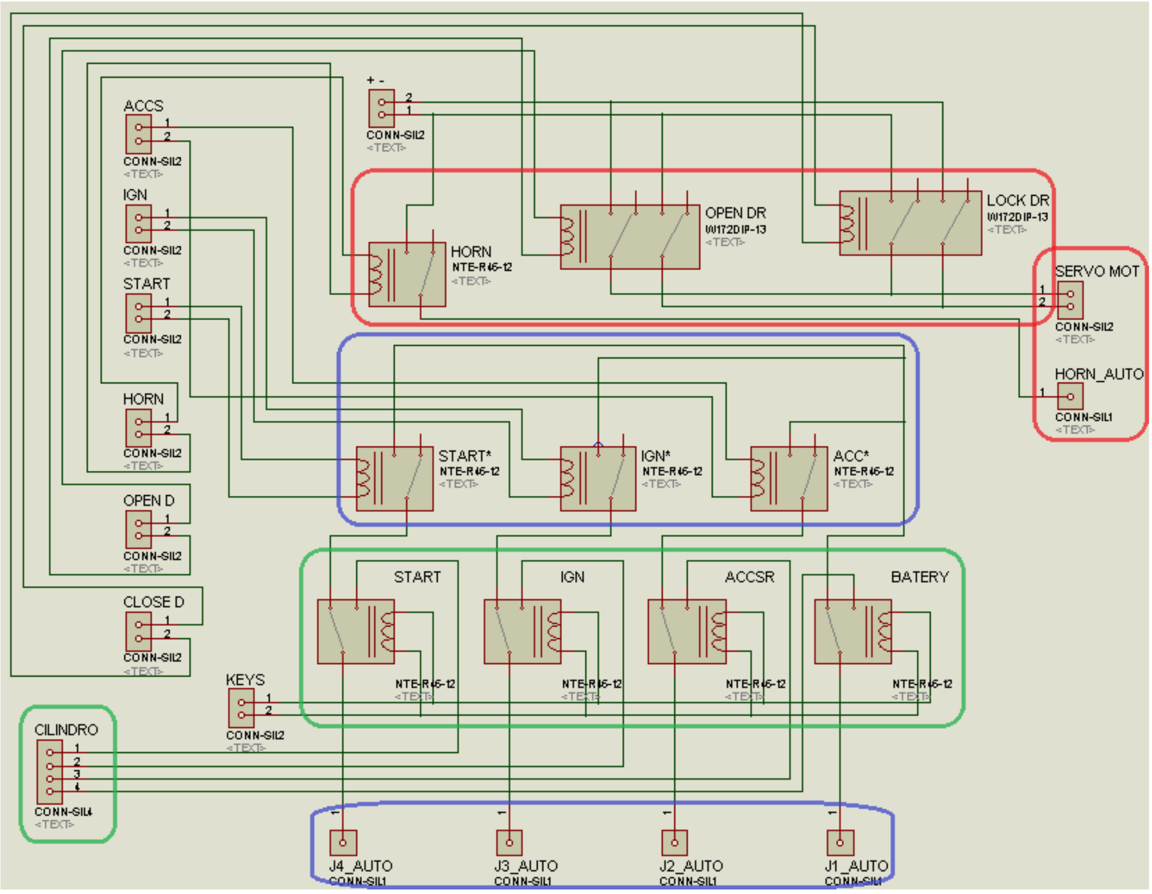


Figura 55. Diagrama de conexiones relés y funciones del automóvil.

Los recuadros pintados en rojo son los relés que controlan los servomotores y la bocina del automóvil. Cuando el usuario desee abrir las puertas, el microcontrolador activa los relés en sincronización: activa dos veces la bocina y después activa el relé que abre la cerradura. Para el caso de poner seguro, la sincronización es hacer sonar la bocina una vez y después activar el servomotor que cierra la cerradura.

Los recuadros pintados en azul, son los relés que están conectados a los cuatro cables que conforman el sistema de arranque del vehículo. Cabe mencionar que el cable “batería” está conectado en común con el recuadro verde. Cuando el usuario desea encender el vehículo, el microcontrolador se encarga de

sincronizar los relés para efectuar el arranque del motor. La sincronización se divide en tres etapas con el siguiente orden, “ON” - “START” - “ON”. Los detalles de estas etapas de encendido se encuentran en la sección 2.1.1. Finalmente, cuando el usuario anhela apagar el vehículo, los relés del recuadro azul se desconectan a través de software del microcontrolador.

Los recuadros verdes, son los relés que están conectados al cilindro de arranque. Éstos controlan la manera en cómo se puede encender el vehículo. Cuando los relés no están activados, sus contactos están en una posición que permite que los relés del cuadro azul controlen el arranque del vehículo, es decir, no es posible encender el auto de la forma tradicional. Cuando el usuario desee encender el vehículo con las llaves, el microcontrolador activa los relés cambiando la posición de los contactores de tal manera que el motor del vehículo se acciona a través del cilindro de arranque y las llaves.

3.9 Relés electromagnéticos para el circuito eléctrico

Los relés electromagnéticos que se utilizan en el sistema de alarma están dimensionados en función a la corriente que deben soportar al momento de controlar el Circuito Primario del sistema de encendido del vehículo. Los detalles del sistema de encendido del vehículo están en la sección 2.1.

La corriente del Circuito Primario se midió directamente de los cables del cilindro de arranque con un multímetro. La corriente que se obtiene es de 5A. El voltaje que necesitan los relés para su funcionamiento es de VDC +12 que se obtiene de la batería del automóvil.

Con los *inputs* listos, se selecciona los relés sobredimensionándolos para mejorar la eficiencia del circuito; por tal motivo, se escoge el relé SONGLE con capacidad de contacto de 15A. Finalmente, el d*atasheet* del relé indica que tiene una potencia de consumo de 0.360mW. Esta característica es uno de los *inputs* que se necesita para seleccionar el transistor del circuito electrónico. Los detalles sobre la selección del transistor que va a controlar a los relés están en la sección 3.7.4.

3.10 Circuito electrónico de prueba

Una vez sustentado el diseño de los segmentos que conforman el circuito electrónico de control (sección 3.4 a la 3.9), se procede a la conexión física de estos elementos electrónicos en una placa de prueba. Este p*rotoboard* (ver Figura 56) permitirá realizar pruebas de conexión sobre el Mini Austin con el objetivo de identificar posibles errores y corregirlos antes de realizar la implementación final sobre placa profesional.

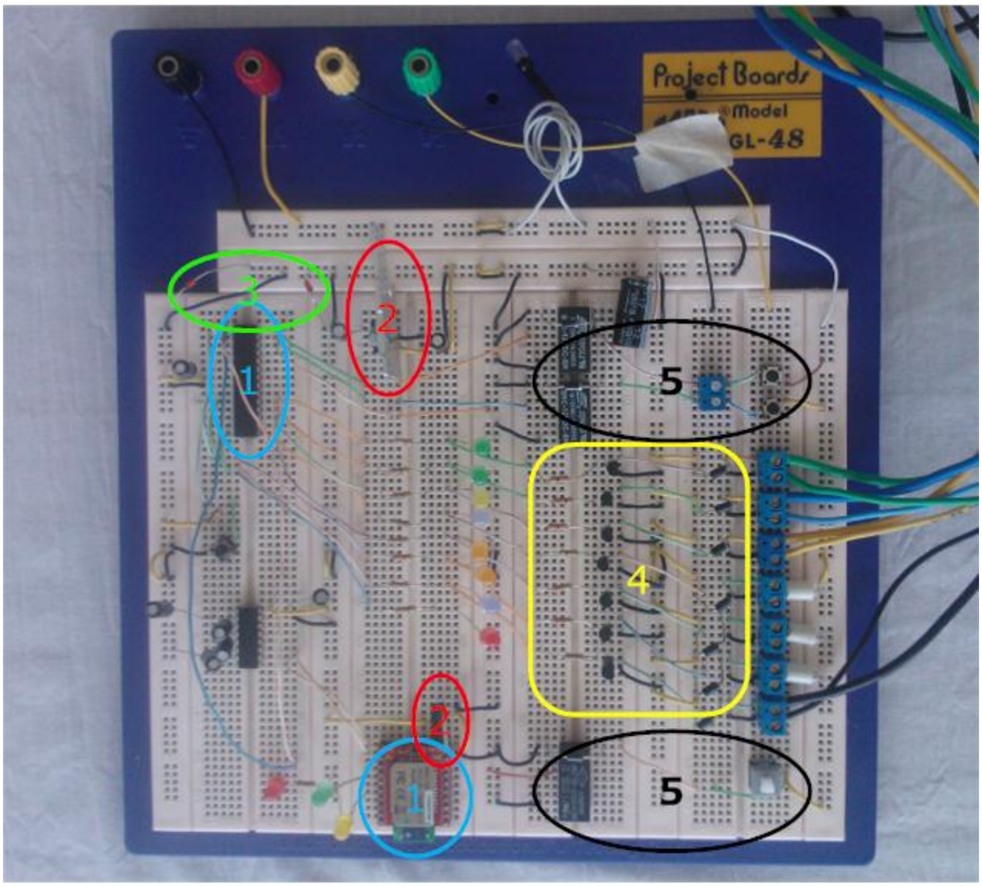


Figura 56. Circuito electrónico de control instalado sobre *protoboard.*

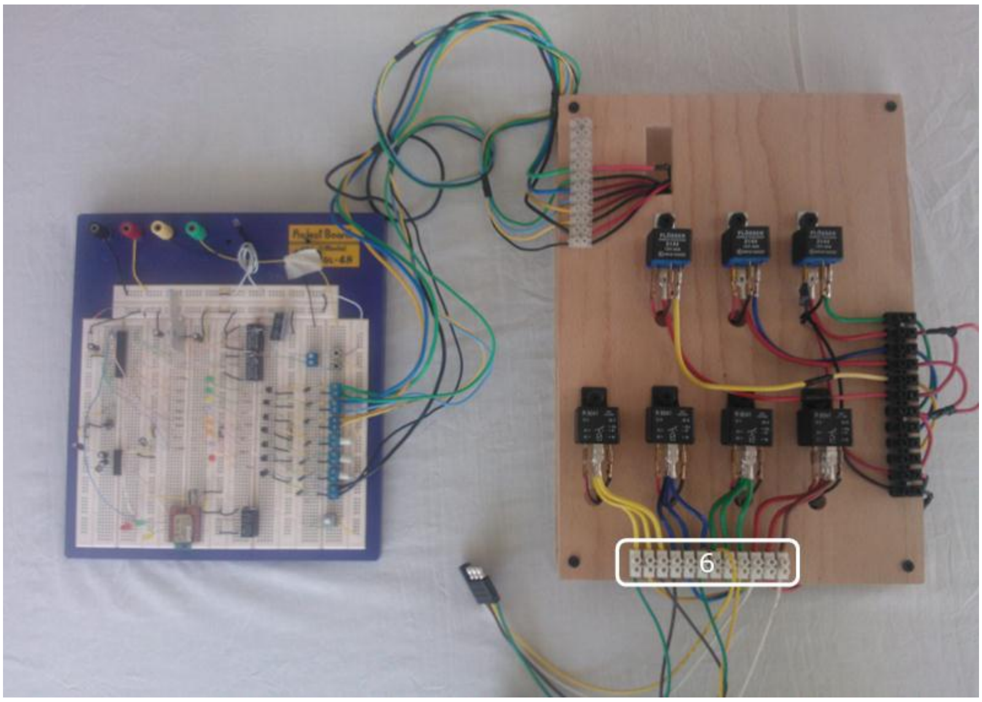


Figura 57. Circuito electrónico de control y el arreglo de relés.

Cada segmento conectado en el *protoboard*, se describe a continuación:

- 1. El módulo Bluetooth y el microcontrolador.
- 2. Los divisores de voltaje.
- 3. Diodos y fusibles de seguridad.
- 4. El arreglo de transistores NPN, los diodos emisores de luz.
- 5. Puertos de entrada.
- 6. Puertos de salida (ver Figura 56 y 57).

3.11 Adecuaciones para el sistema de alarma e inmovilización vehicular

Para completar el diseño del sistema de alarma e inmovilización vehicular planteado en las secciones 3.1 al 3.10, es necesario realizar las adecuaciones electrónicas en el Mini Austin que se describen a continuación:

- Conectar los cables del Circuito Primario y del cilindro del vehículo a los respectivos puertos de salida del circuito de control (ver Figura 62).
- Instalar diodos de protección en los pulsadores y botones del vehículo para evitar inducciones que perjudiquen el funcionamiento de los pines de entrada del circuito (ver Figura 63).
- Instalar servomotores para la apertura y cierre de la cerradura de las puertas (ver Figura 64).
- Realizar una conexión en paralelo al bulbo del freno para conectarlo al pin de entrada (ver Figura 65).
- Realizar una conexión en paralelo a la bocina del vehículo para conectarlo al pin de salida (ver Figura 66).



Figura 58. Conexión de cables: Circuito Primario y cilindro del vehículo.



Figura 59. Diodo de protección en el botón del vehículo.



Figura 60. Servomotor instalado en la puerta del vehículo.

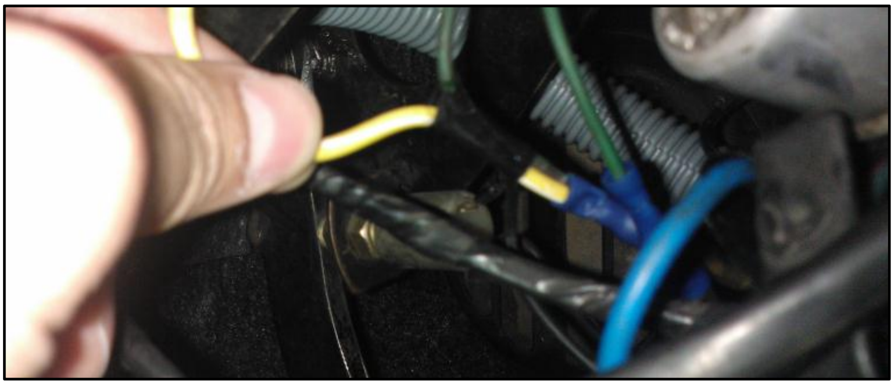


Figura 61. Conexión en paralelo al bulbo del freno.

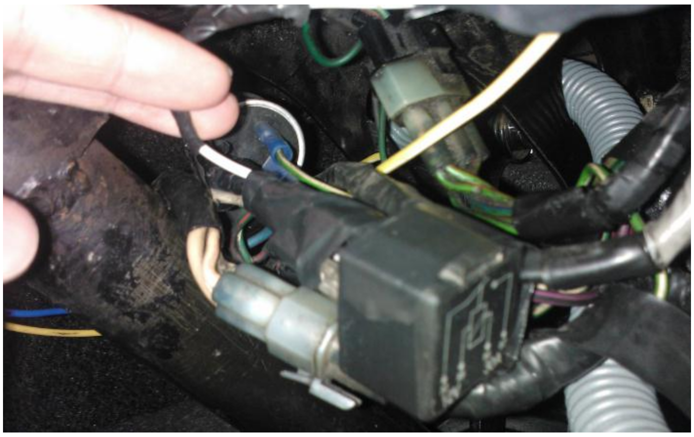


Figura 62. Conexión en paralelo al relé de la bocina del vehículo.

CAPÍTULO 4

4. Pruebas y Resultados

La presentación de las pruebas y resultados obtenidos durante la implementación total del sistema de alarma e inmovilización vehicular se dividen en tres secciones: la primera presenta los resultados obtenidos durante las pruebas realizadas en la implementación del circuito electrónico de control, la segunda despliega los resultados obtenidos tras verificar el funcionamiento de la aplicación Interface Gráfica de Usuario instalada en el *smartphone* y la tercera presenta los resultados obtenidos en las pruebas de conexión entre el teléfono inteligente y el módulo Bluetooth.

4.1 Pruebas y Resultados en la implementación del circuito electrónico de control

Recapitulando las secciones que conforman las etapas de diseño e implementación del circuito electrónico de control:

- La programación de la aplicación en Visual Studio (sección 3.1).
- La configuración de las propiedades para del puerto serial y del Bluetooth en el teléfono inteligente (sección 3.2).
- La ejecución de la aplicación Interface Gráfica de Usuario en el smartphone (sección 3.3).
- La configuración del Módulo Bluetooth RN-41 (sección 3.4).
- La implementación del programa en el microcontrolador incluyendo sus respectivas simulaciones en Proteus (sección 3.5, 3.6 y 3.7).
- El ensamblaje de los segmentos electrónicos en el *protoboard* (sección 3.8, 3.9 y 3.10).

- El acondicionamiento electrónico en el Mini Austin (sección 3.11).

Una vez concluida estas secciones, se procede a montar el proyecto en el vehículo para verificar su correcto funcionamiento y corregir posibles fallas que puedan surgir antes de construir el circuito Impreso definitivo (ver Figura 63).

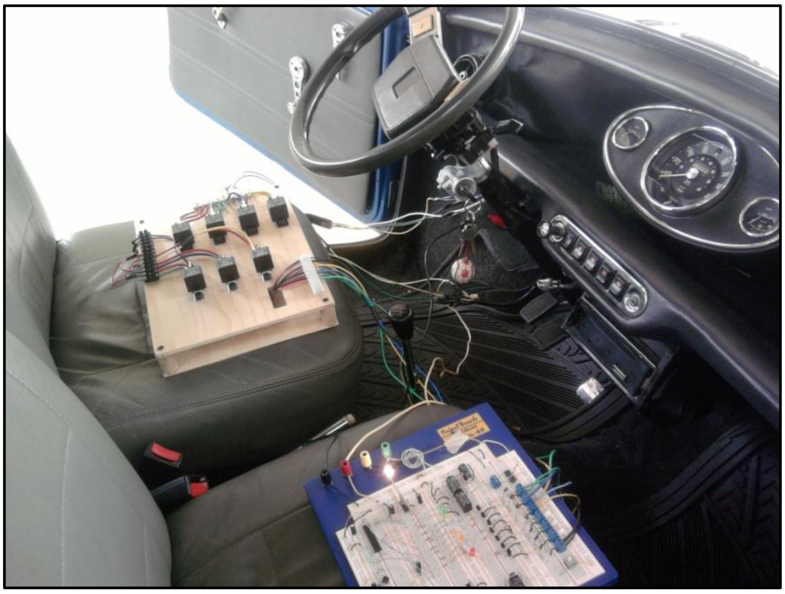


Figura 63. Pruebas realizadas sobre Mini Austin con el circuito electrónico de prueba.

El siguiente video presenta una prueba de arranque cuyo objetivo es evidenciar el encendido del motor con el sistema en desarrollo; sin embargo, la prueba revela un desperfecto: un error en la conexión del *smartphone* y el módulo Bluetooth. La solución a este inconveniente llevó a fortalecer la programación del celular para obtener una conexión autónoma eficiente entre el *smartphone* y el módulo Bluetooth (ver Video 9).

http://youtu.be/-q_8kEcMT7w

Video 9. Prueba de encendido del vehículo a través del botón “ONEngine”.

Una vez comprobado que la aplicación del *smartphone*, la conexión Bluetooth, el microcontrolador y los elementos electrónicos del circuito de control trabajan en conjunto sin reportar errores, se procede a diseñar y fabricar el circuito electrónico de control en placa profesional. El layout del circuito se indica a continuación (ver Figura 64 y 65).

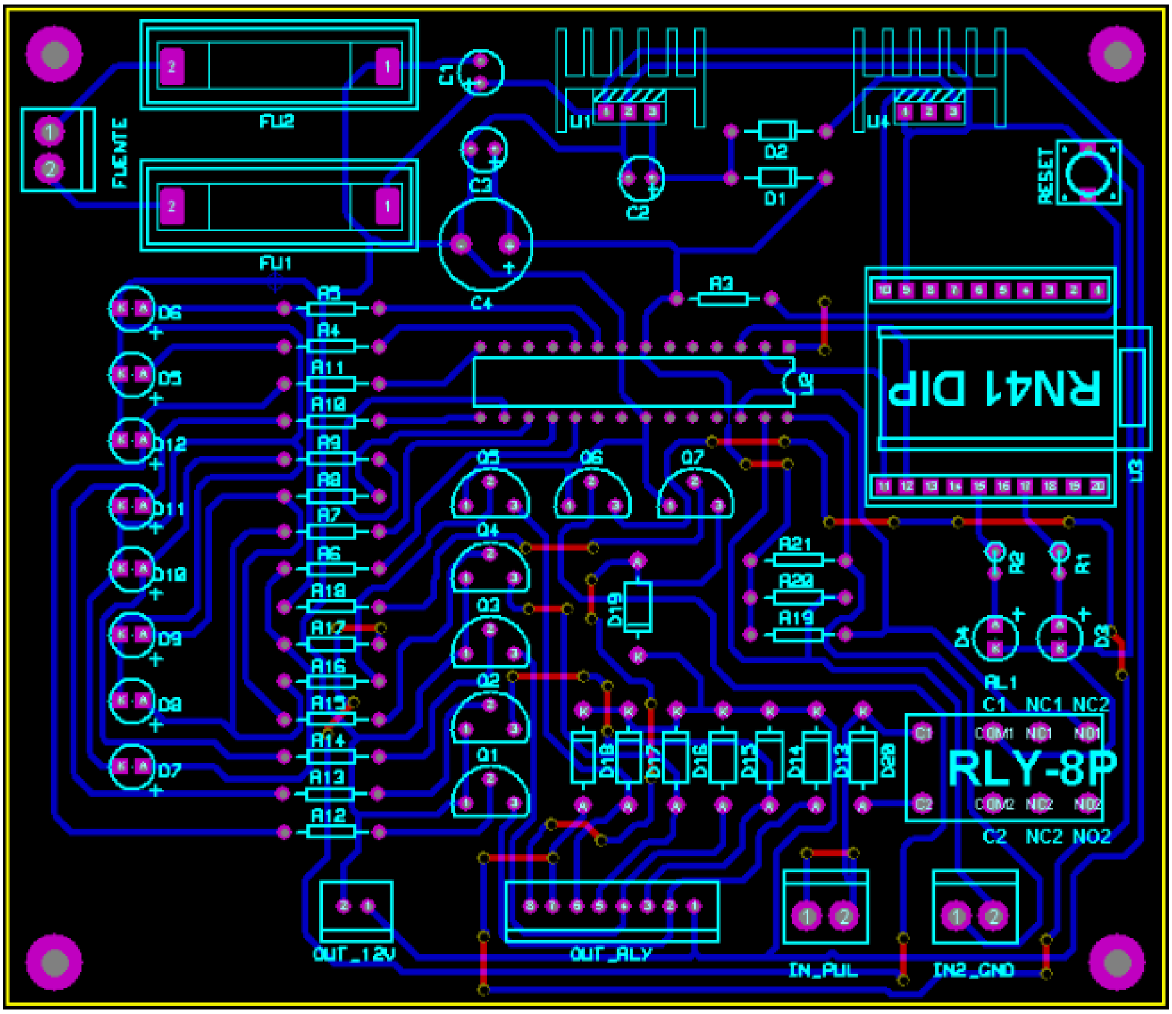


Figura 64. Layout circuito de electrónico de control.

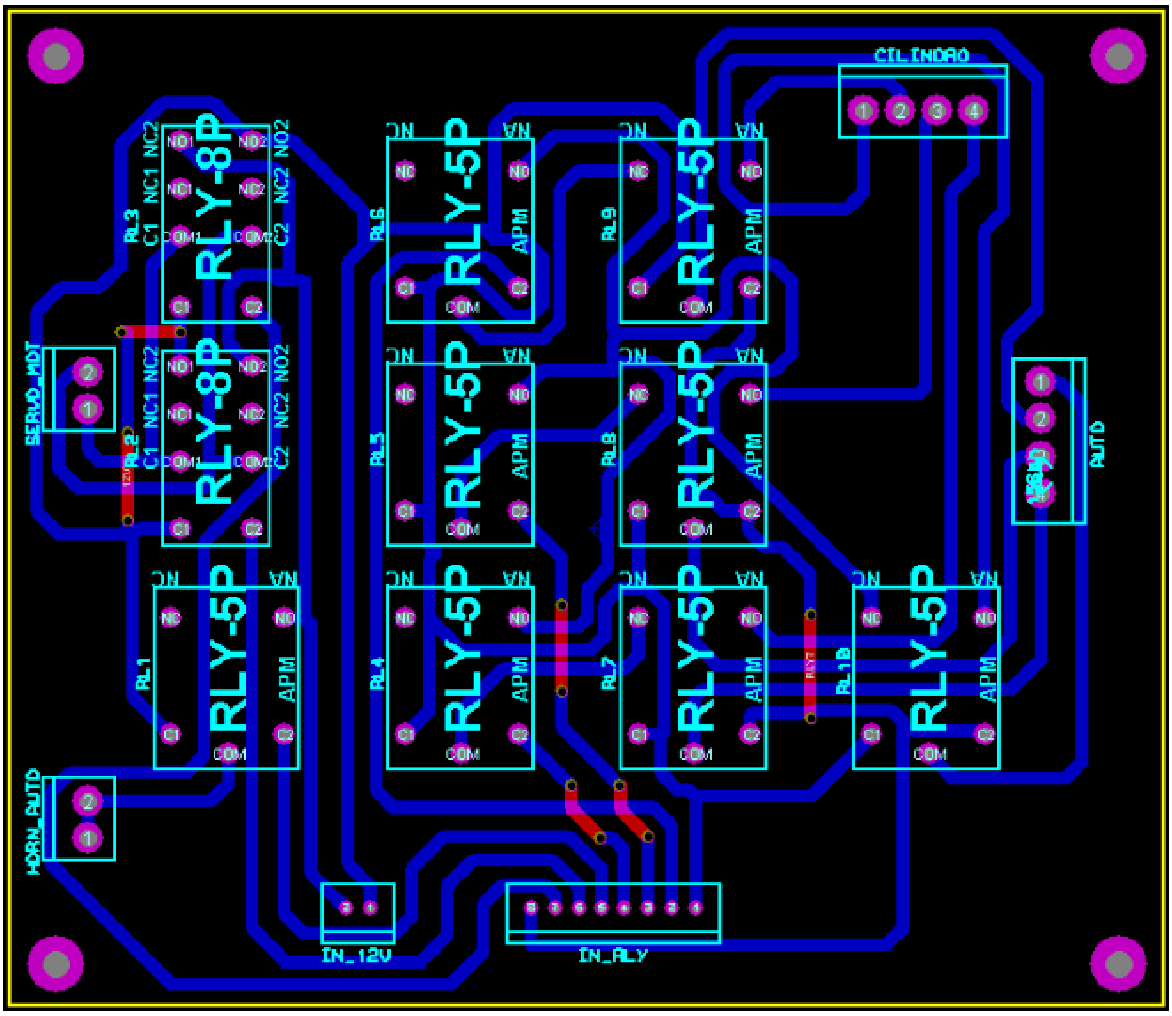


Figura 65. Layout circuito electrónico de control, arreglo de relés.

El circuito electrónico de control se conforma de los siguientes elementos:

- 1. El Módulo Bluetooth y el microcontrolador.
- 2. Los divisores de voltaje.
- 3. Diodos y fusibles de seguridad.
- 4. El arreglo de transistores NPN, los diodos emisores de luz y el arreglo de relés electromagnéticos.
- 5. Puertos de entrada.
- 6. Puertos de salida (ver Figura 66).

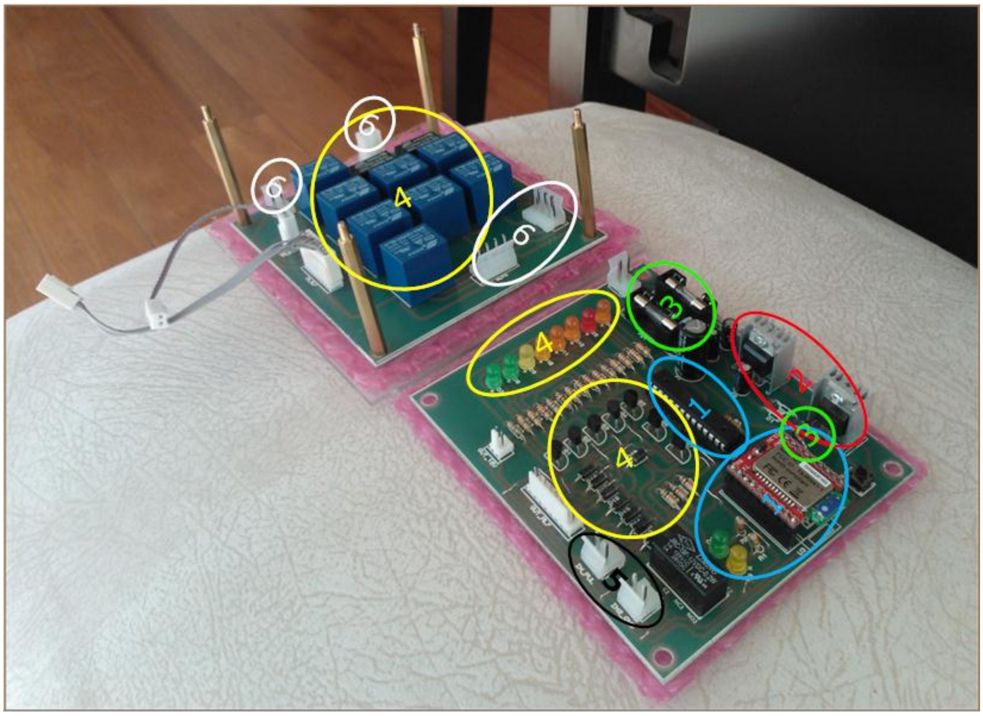


Figura 66. Circuito electrónico de control final extendido.

Para combatir los factores climáticos como la humedad y el polvo, el circuito electrónico se construye en placa profesional verde y se protege las soldaduras con empaque de burbujas (ver Figura 67).

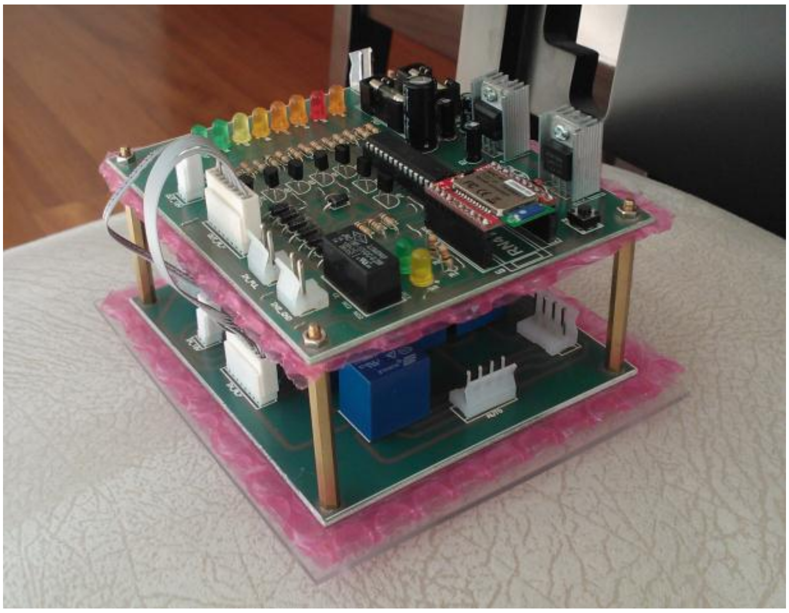


Figura 67. Circuito electrónico de control final compacto.

Para que el dispositivo pueda caber debajo del tablero del vehículo, el circuito electrónico final se implementa en dos placas conectadas entre sí obteniendo un dispositivo con las siguientes dimensiones (13 x 11 x 9) cm (ver figura 68).

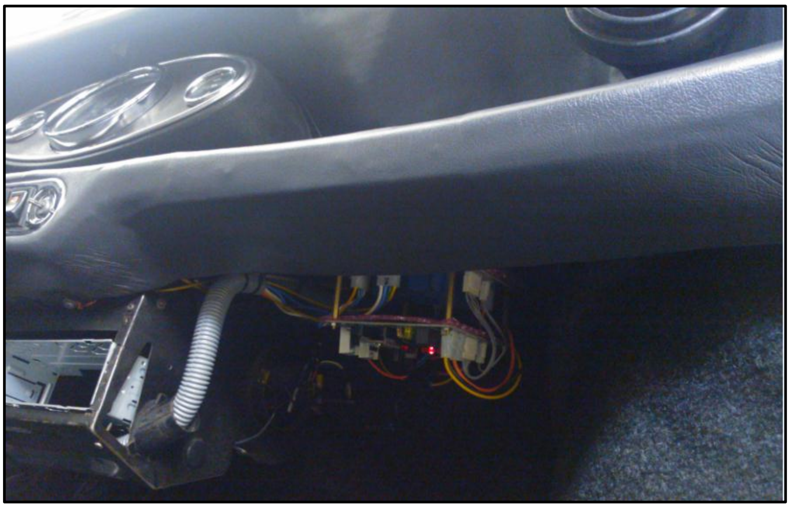


Figura 68. Ubicación del circuito de control en el Mini Austin.

Se mide con un multímetro el voltaje y la corriente del dispositivo para verificar su potencia de consumo mientras espera establecer una conexión Bluetooth. Las mediaciones obtenidas son 5V y 0.04A, lo que corresponde un consumo de 0.2W.

4.2 Pruebas y resultados sobre la Interface Gráfica de Usuario en el *smartphone*

Los resultados de las pruebas realizadas sobre las funciones que posee la Aplicación programada en el smartphone se presentan en las siguientes tablas y videos (ver Tablas 7 y 8, Videos 10 al 18).

Aplicación Interface Gráfica de Usuario en el *smartphone*					
Ubicación Original	**Subpanel**	**Función/es**	**Descripción**	**Acción**	**Video**
Panel Principal	**Off Engine**	"Off Engine"	Apagar el motor	Envía "o" a través de Bluetooth	Video 10
	On Engine	"START"	Encender el motor	Envía "e" para empezar el arranque del motor y "r" para detener el mismo.	Video 11
	Accessories	"Accessories"	Controlar Accesorios	Envía "z" para activar o desactivar Accesorios	Video 12
	Doors	"Open" Door	Abrir seguro de la Puerta	Envía "a" para activar el servomotor y quitar el seguro de la puerta	Video 13
		"Lock" Door	Poner Seguro a la Puerta	Envía "c" para activar el servomotor y poner seguro de la puerta	
Menu	**Start**	-	Acceso al Panel Principal	Una vez ingresado correctamente la clave, "Start" se deshabilita	
	Password Setting	-	Cambiar la clave de Ingreso a la aplicación del *smartphone*	1) Comprobar el usuario verdadero, solicitando la antigua clave	Video 14
				1a) Si no ingresa correctamente, seguirá solicitando infinitamente la antigua clave	
				2) Ingresada correctamente la antigua clave, se solicita que escriba la nueva clave por dos ocasiones y que sea mayor a 4 caracteres	
				2a) Si no ingresa correctamente la nueva clave, aparece mensaje de "Mismatched Password"	
				3) Ingresada correctamente la nueva clave, se la guarda de manera encriptada y se retorna al inicio de subpanel Password Setting	
	Car Keys Settings	"w/ Keys"	Activa el cilindro de las llaves	Envía "k" para activar la operatividad del cilindro	Video 15
		"w/o Keys"	Desactiva el cilindro de las llaves	Envía "l" para desactivar la operatividad del cilindro	
	Exit	-	Salir de la Aplicación	Salir del programa y cierra la conexión Bluetooth	

Tabla 7. Subpaneles de la aplicación instalada en el *smartphone.*

http://youtu.be/MO0H-8jm1qk

Video 10. Función “Off Engine” de la aplicación Visual Studio.

http://youtu.be/Dn74hceqJIY

Video 11. Función “On Engine” de la aplicación Visual Studio.

http://youtu.be/9vL6ZVxTCKw

Video 12. Función “Accessories” de la aplicación Visual Studio.

http://youtu.be/mDsqRtM1cAo

Video 13. Función “Open” & “Lock” Door de la aplicación Visual Studio.

http://youtu.be/IU2OlZlWRUg

Video 14. Subpanel “Password Setting” de la aplicación Visual Studio.

http://youtu.be/k4ehdy0hBWY

Video 15. Funciones “w/ & w/o Keys” de la aplicación Visual Studio.

Funciones de la Interface Gráfica de Usuario en el *smartphone*				
Tipo de Función	**Funciones Especiales**	**Descripción**	**Acción**	**Video**
Seguridad	Revisar si conexión Bluetooth está establecida para encender el vehículo	Cuando se acciona el botón ONEngine en el panel del auto, el Smartphone recibe el dato "m" y contesta inmediatamente con "n" a través del puerto COM2	1). Si es exitoso el proceso de recibir y enviar "m" y "n", el encendido de motor se lleva a cabo 2). No es exitoso el proceso de recibir y enviar "m" y "n", el µC acciona el Pito 3 ocasiones	Video 16
Protocolo de encendido del vehículo	Informar la acción de aplastar el pedal del freno	El µC a través del Smartphone informa al usuario que necesita aplastar el freno para encender el vehículo	Mostrar un mensaje en la pantalla del Smartphone, indicando que se necesita accionar el pedal del freno	Video 17
Seguridad	Cifrado de Claves	Para leer y guardar la clave de acceso a la aplicación, se instauro un sistema de encriptación	Para leer o guardar la clave, se corre varias veces la función de encriptado MD5	Video 18

Tabla 8. Funciones especiales de la aplicación instalada en el *smartphone.*

http://youtu.be/CMoWSPMrDMU

Video 16. Función especial, conexión Bluetooth establecida para encender el vehículo.

http://youtu.be/Y7-zXcJuCDU

Video 17. Función especial, informar la acción de presionar el pedal del freno.

http://youtu.be/fC0n_YGrqmA

Video 18. Función especial, cifrado de claves.

4.3 Pruebas y Resultados en la conexión entre el smartphone y el módulo Bluetooth

Las pruebas están enfocadas a verificar las características de autonomía y seguridad que posee la conexión Bluetooth entre el *smartphone* y el módulo Bluetooth RN-41.

La tecnología Bluetooth trabaja en los tres primeros niveles de la arquitectura OSI (Físico, Enlace y Red), el tercer nivel es el responsable del establecimiento y la finalización de las conexiones. Con este fundamento teórico puesto en práctica, la aplicación en el *smartphone* fue diseñada de tal manera que el usuario no tiene que preocuparse por revisar si la conexión está establecida cada vez que desee enviar instrucciones al microcontrolador. Para lograr esta autonomía, cada función de la aplicación fue diseñada de tal manera que, antes de ejecutar su tarea asignada debe revisar si la conexión Bluetooth está establecida o no. En el caso de que sí exista, la función simplemente ejecuta su acción, pero en el caso que no exista, entra en un proceso de verificación de usuario e intenta establecer la conexión con el módulo Bluetooth (ver Video 19).

http://youtu.be/pFTZFc5s6yU

Video 19. Prueba para verificar que la conexión Bluetooth es autónoma.

Las características de seguridad que brinda el Módulo Bluetooth RN-41 son: activar o desactivar los *features Bonding* e *Invisible* y cambiar el *Pin code* por un *string* más complejo y extenso.

Cuando el *feature Bonding* está activado, el Módulo RN-41 solo acepta conexiones con dispositivos cuya dirección Bluetooth se encuentre registrada y certificada anteriormente. Esta opción no fue empleada en la implementación final del proyecto por motivos de soporte remoto y mantenimiento, sin embargo, el video a continuación indica la manera en que se implementa dicha característica en el Módulo RN-41 (ver Video 20).

http://youtu.be/m4-TI2thmmc

Video 20. Prueba para verificar *feature Bonding* en el módulo RN-41.

Cuando el *feature Invisible* está activado, el módulo RN-41 evita que dispositivos remotos establezcan conexiones ya que éste se encuentra oculto; sin embargo, aquellos equipos que poseen el *Pin code* pueden establecer una conexión con el módulo. Esta opción de seguridad, a diferencia de *Bonding*, sí fue utilizada en la implementación final del proyecto. Las pruebas y resultados de esta implementación se presentan en el video a continuación (ver Video 21).

http://youtu.be/0b_tmW8-7-4

Video 21. Prueba para verificar *feature Invisble* en el módulo RN-41.

Finalmente, como se indicó en la sección 3.4, el *Pin code* del módulo fue cambiado por uno nuevo con una extensión de 6 caracteres. Se evaluó la opción de extender la longitud del *Pin code*; sin embargo, no se aplicó dicho cambio porque la conexión entre el *smartphone* y el módulo RN-41 tendía a generar errores de emparejamiento.

CAPÍTULO 5

5. Conclusiones y Recomendaciones

5.1 Conclusiones

El objetivo principal del proyecto, el cual consiste en diseñar e implementar un sistema de alarma e inmovilizador vehicular controlado por un teléfono inteligente a través de conexión Bluetooth, se cumplió con éxito. Los objetivos específicos, los cuales definieron los estándares de calidad a obtenerse en este proyecto, también fueron logrados en su totalidad. A continuación, se presentan los argumentos que certifican el cumplimiento de todos los objetivos propuestos en esta tesis.

1). La primera sección del proyecto estipuló crear una aplicación *smartphone* intuitiva e invulnerable.

Los múltiples videos realizados en la sección 4.2 evidencian que la aplicación hecha en Visual Studio 2008 sobre la plataforma Windows Mobile 6.5 es intuitiva y fácil de manejar (ver videos 10 al 15).

El objetivo de desarrollar una aplicación invulnerable se obtiene con el proceso de cifrado MD5 utilizado al momento de leer y grabar la clave de acceso de la aplicación. El video de la sección 4.2 indica los detalles de este proceso implementado (ver video 18).

2). La segunda sección del proyecto determina establecer una conexión Bluetooth autónoma e indescifrable entre el *smartphone* y el microcontrolador.

La conexión autónoma se consigue eficazmente porque el usuario no tiene que revisar, si la conexión Bluetooth está hecha cada vez que desee controlar la alarma vehicular, tampoco tiene que reiniciar el equipo para levantar la conexión en el caso que no esté hecha. Las justificaciones a estos argumentos indicados se encuentran en la sección de diseño 3.1, en la sección de pruebas y resultados 4.3 y en el video 19 "Conexión Bluetooth autónoma".

La conexión indescifrable se logra de manera exitosa a través de los *features* "*Invisible*" y "*Pin code* complejo" implementados en el Módulo Bluetooth RN-41. Los detalles sobre las capacidades, el funcionamiento y la implementación de estos *features* se presentan en la sección 4.3; sin embargo, cabe mencionar que, el *feature Bonding* no es implementado en el proyecto por motivos de brindar un soporte remoto al usuario en el caso que llegue a fallar el Bluetooth del smartphone.

3). La tercera sección señala la implementación de un microcontrolador y un circuito electrónico con las siguientes características: multitarea, de bajo consumo eléctrico, compacto y resistente a factores climáticos.

La función de multitarea es constatada por las capacidades de velocidad y cantidad de pines que posee el microcontrolador ATMEGA8. Con una velocidad de 1 MIPS (Millón de Instrucciones por segundo), tiene capacidad suficiente para ejecutar todas las tareas de entrada y salida de manera cuasi-simultánea. Adicionalmente, el sistema de alarma emplea 14 de los 23 pines programables

que posee el ATMEGA8. Estas dos características en conjunto permitieron obtener la función de multitarea en el circuito electrónico de control.

Se cumple con la condición de ser un dispositivo de bajo consumo eléctrico, porque el circuito consume 0.2W de la batería del vehículo mientras espera conectarse al *smartphone*, dicho argumento es mencionado en la sección 4.1.

La condición de ser compacta se obtiene al implementar el circuito final sobre dos placas montadas entre sí. El tamaño de este dispositivo es lo suficientemente pequeño para caber debajo del tablero del Mini Austin como se indica la figura 61 de la sección 4.1.

Para que el circuito de control sea resistente a los factores de humedad, temperatura y polvo, se implementa el dispositivo en placa profesional y se protegen las soldaduras con empaque de burbujas como se indica en la figura 65 de la sección 4.1

4). Basándose en los resultados obtenidos en las pruebas y en los objetivos propuestos, el sistema de alarma e inmovilizador vehicular responde satisfactoriamente en términos de calidad e innovación.

5.2 Recomendaciones y expectativas comerciales

El proyecto desarrollado puede ser el punto de partida para promover nuevas tesis cuyas propuestas sean implementar sistemas de alarma e inmovilización vehicular controlados por *smartphone*. Las nuevas tesis podrían apoyarse y utilizar el microcontrolador, el módulo Bluetooth y el circuito de control implementado en este proyecto; sin embargo, el *smartphone* a utilizarse se recomienda que sea de diferente sistema operativo al Windows Mobile 6.5 con el objetivo de expandir y llegar a más usuarios que poseen *smartphones* como Blackberry, iPhone y Android.

La otra recomendación, en el caso que un estudiante desee realizar una tesis similar, sería que trate de implementar el proyecto en un automóvil con un sistema electrónico más moderno con la finalidad de obtener una señal de control fiable que le indique si el motor del vehículo está encendido o no. Con esta ventaja electrónica, el nuevo proyecto podría implementarse con un sistema de arranque completamente automático y autónomo.

Finalmente, son varias las visiones económicas que puede tener esta tesis, entre ellas: lanzarla como una nueva opción comercial en el mercado actual de las alarmas vehiculares, instalar el proyecto en una flota de transporte público de tal manera que cada chofer pueda ingresar y encender el vehículo solamente utilizando su celular; y finalmente, implementar el proyecto en una empresa de renta de vehículos. La idea de esta última es que el usuario rente por internet y después se dirija directamente al vehículo para utilizarlo. Todo esto es posible, porque después que el usuario concrete el arrendamiento del vehículo, la

aplicación de control de la alarma del vehículo arrendado se instala automáticamente en su *smartphone*. Con esta alternativa de servicio, el usuario podría retirar y dejar el vehículo en múltiples zonas de una manera muy flexible porque ya no tiene la necesidad de pasar por una oficina y retirar las llaves del vehículo.

BIBLIOGRAFÍA

Abu-Ein, Qusai, y otros. «Automobile Functions Control System Via Bluetooth Utility.» *Australian Journal Of Basic & Applied Sciences* 3.4 (2009): 3273-3282. Academic Search Complete. 19 de Sept. de 2012.

Anderson, Scott. «Electronic Engineering Times.» *Academic OneFile* 5 de Mar de 2001: 93. Web, EBSCO.

Arrayás, Manuel. *Electromagnetismo, circuitos y semiconductores.* España: Dykinson, 2008. Web ebrary.

Atmel Corporation. *Atmega8 Darasheet*. San Jose, CA, 2011. Electronic Source.

Billiet, Walter. *Entretenimiento y reparaci[on de motores de automóvil*. España: Editorial Reverté S.A., 1979. Físico Impreso.

Brooks, R. A. «A robust layered control system for a mobile robot.» *IEEE Robotics and Automation* RA.2 (1986): 14. EBSCO.

Carmody, Thomas. «Bluetooth Paves The Way For Truly Wireless Car Interiors.» *ECN: Electronic Component News* Feb de 2007: 28-30. Web.

Cheng Yu, Jade. *jade-cheng.com*. s.f. Web. 1 de Oct de 2012.

Cisilino, Mario. *Reparando la electrónica de la línea blanca*. Argentina: Editorial Hispano America HASA, 2009. Web ebrary.

El Comercio. *El robo de vehículos registró un incremento en el año que termina*. Quito: Javier Ortega, 12 de Diciembre de 2013. WEB. <http://www.elcomercio.ec/seguridad/Quito-inseguridad-robo_de_vehiculos_0_838116266.html>.

García Breijo, Eduardo. *Compilador C CSS y Simulador PROTEUS para Microcontroladores PIC*. Barcelona: Marcombo, Ediciones Técnicas, 2008. Web.

—. *Compilador C CSS y Simulador PROTEUS para Microcontroladores PIC*. Vol. VII. Barcelona: Marcombo, Ediciones Técnicas, 2008. Web ebrary.

Gattiker, Urs E. *The Informatio Security Dictionary*. Boston: Kluwer Academic Publishers, 2004. eBook.

Harris, Tom. «How Car Alarms Work.» 02 de August de 2001. *auto.howstuffworks.com.* Web. 30 de October de 2012. <http://auto.howstuffworks.com/car-alarm.htm>.

Informer Technologies. «software.informer.com.» 2011. *Bascom-AVR.* Web. 2 de Oct de 2012. <http://bascom-avr.software.informer.com/>.

Krenik, Bill y Mike Yonker. «EE Times Home.» 4 de 16 de 2004. *News and Analysis.* Texas Instruments. Web. 2012 de Oct. de 1. <http://i.cmpnet.com/commsdesign/2004/cellwifi-fig2.jpg>.

Larousse Editorial, S.L. «relé s. m.» *Diccionario Manual de la Lengua Española*. Larousse Editorial, S.L., 2007. WEB. <http://es.thefreedictionary.com/rel%C3%A9>.

Majon International. «Majon.com.» 24 de Oct de 2009. *What is TTL and How Does it Relate to RS232.* Ed. Chris Robertson. Web. 1 de Oct de 2012. <http://ezinearticles.com/?What-is-TTL-and-How-Does-it-Relate-to-RS232?&id=3147872>.

Microsoft. «Smart Device Development.» 2012. *msdn.microsoft.com.* Web Page. 19 de Sept. de 2012. <http://msdn.microsoft.com/en-us/library/sa69he4t(v=vs.90).aspx>.

Motsay, Emily. «Bluetooth Holds Telematics Promise.» *RCR Wireless News* 21.43 (2002): 8. EBSCO.

Osborne, Adam. «An Introduction to Microconputers Volume 1: Basi Concepts.» 1980. *Wikipedia.org.* McGraw Hill. Web. 1 de Oct de 2012.

processors.wiki.ti. «Texas Instruments.» 2 de March de 2012. *StarterWare UART/IrDA/CIR.* Web. 1 de Oct de 2012. <http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter#cite_note-Osborne80-0>.

Read, P.P.J y V.C Reid. *Manual Técnico del Automóvil*. Madrid: A. Madrid Vicente, Ediciones, 2001. Físico.

Rodríguez, Carlos A. *Curso básico de electrónica*. Argentina: El Cid Editor, 2007. Web ebrary.

Roving Networks. «Advanced User Manual.» *Roving Networks Bluetooth Product User Manual*. Version 4.77. 29 de August de 2009. WEB.

—. *RN_41*. 1 de Oct de 2012. Web. 1 de Oct de 2012.

—. «RN-41/RN-41-N Class 1 Bluetooth Module Datasheet.» *RN-41/RN-41-N Class 1 Bluetooth Module*. Los Gatos, CA: Roving Networks Inc., 2012. PDF.

Semtech Electronics Ltd. *ST 2SC1383 Datasheet*. Hong Kong, 7 de Dic de 2002. Web Datasheet.

—. *ST 7805 Datasheet*. Hong Kong, 21 de Jun de 2006. Web.

smart-GSM. *smart-gsm.com*. 2012. Web Page. 19 de Sept. de 2012. <http://www.smart-gsm.com/moviles/samsung-i637-jack>.

Sparkfun. *Beginning Embedded Electronics - 2*. 19 de June de 2008. WEB. 22 de Febrero de 2013. <http://www.sparkfun.com/tutorials/93>.

st2.gsmarena. *st2.gsmarena.com*. 2012. Web Photo. 19 de Sept. de 2012. <http://st2.gsmarena.com/vv/pics/samsung/samsung-i637-jack-1.jpg>.

TextosCientificos. *TextosCientificos.com*. 25 de Jul de 2005. Web. 1 de Oct de 2012. <http://www.textoscientificos.com/redes/comunicaciones/modos>.

Thoone, Martin. «Electronics Drive Innovation, Enhancing The Driving Experience.» *Ward´s Auto World 40* (2004): 68. Business Source Premier. 18 de Sept. de 2012.

uControl. «Fundamentos de la Transmisión Sincrónica.» *uControl* 1 de Ene de 2008: 8-15. Web.

Williamson, Jason. «"{10 Golden Rules For HMI Design}".» *Appliance Design* 58.11 (2010): 14-16. Business Source Premier.

Willydev. «Boletín Febrero 2008.» 17 de Feb de 2008. *willydev.net.* Web Page. 19 de Sept de 2012. <http://www.willydev.net/InsiteCreation/v1.0/willycrawler/2008.05.01.articulo.lo%20nuevo%20en%20visual%20studio%202008.pdf>.

ANEXOS

Código implementado en el microcontrolador ATMEGA8

```
$regfile = "m8def.dat"
$crystal = 8000000
$baud = 9600

$hwstack = 40
$swstack = 40
$framesize = 40


Ddrb = 255 : Portb = 0                 'Config PortB = Output
Ddrc = 0 : Portc = 255                 'Config PortB = Input "acciona con bajos"


Dim Serialchar As Byte
Dim Ischar As Byte
Dim Bphoneok As String * 3

Dim Cprevburn As Single
Dim Fbuffer As Byte
Dim Ccheckphone As Single
Dim Facesorio As Byte
Dim Chorn As Byte



Main:

Portb.7 = 1                            'bandera pa saber que sucedio el evento OFFENGINE

Cprevburn = 0                          'Reset controlador que evita quemar Motor de
Arranque
Fbuffer = 0
Ccheckphone = 0
Facesorio = 0

Do
  Serialchar = Inkey()                 'la clave de todo este codigo, Inkey()

  Select Case Serialchar
    Case &H61                          'ident. letra "a" Door Open
      Gosub Dooropen

    Case &H63                          'ident. letra "c" Door Close
      Gosub Doorclose

    Case &H65                          'letra "e" Enceder el motor
      Portb.5 = 0                      'Desactivando Accesorios
      Portb.4 = 1                      'Encendiendo ING
      Wait 2
      Portb.3 = 1                      'Encendiendo START
      Gosub Startphone
```

```
    Case &H6F
      Goto Offengine                      'H6F = "o" OFF

     Case &H7A                            'H78 = "z"
      Gosub Accessories

     Case &H6B                            'letra "k" para activar encendido con llaves
      Gosub Keyfull
     Case &H6C                            'letra "l" para apagar encendido con llaves
      Gosub Keyless
    End Select


     If Pinc.5 = 0 Then                   'boton de apagado, =0 significa que esta aplastado
      Goto Offengine
     End If

     If Pinc.4 = 0 Then                   'Ident. Encendido manual
      Waitms 300                         ' Encender EngineLED
      Gosub Prestart
     End If
Loop
End                                  'end program



Dooropen:
  Portb.2 = 1 : Waitms 50                    'Bocina 1re Click
  Portb.2 = 0 : Waitms 130
  Portb.2 = 1 : Waitms 50                    'Bocina 2do Click
  Portb.2 = 0 : Wait 2

  Portb.1 = 1                            'Mecanismo de abrir puerta
  Waitms 40                               'Mecanismo de abrir puerta
  Portb.1 = 0                            'Mecanismo de abrir puerta
Return

Doorclose:
  Portb.2 = 1 : Waitms 50                    'Bocina 1re Click
  Portb.2 = 0 : Wait 2

  Portb.0 = 1                            'Mecanismo de cerrar puerta
  Waitms 70                               'Mecanismo de cerrar puerta
  Portb.0 = 0                            'Mecanismo de cerrar puerta
Return



Prestart:                              ' PressStart, reviso la presencia o no, del celular
  Portb.5 = 0                            'Se desactiva Accesorios
  Input "M"                               'Envio un Caracter "M" para que el celular me responda
"n"
  Do
    Fbuffer = Ischarwaiting()
    If Fbuffer = 1 Then
```

```
        Bphoneok = Inkey()

        If Bphoneok = "n" Then                    ' Si recibo "n", indica que Celular esta activo.
          Goto Prestart1
        End If
      Else

        Incr Ccheckphone
      End If
    Loop Until Ccheckphone > 15000                ' Si no recibo confirmacion del Celular, bocina
                                                       de auto se activa

    Fbuffer = 0
    Ccheckphone = 0
    For Chorn = 1 To 3
    Portb.2 = 1 : Wait 1                         'Momento donde la bocina se activa fuertemente
    Portb.2 = 0 : Wait 1
    Next
    Wait 1

  Return


  Prestart1:                                  ' Reviso si esta o no el Freno Pisado,
    Fbuffer = 0
    Ccheckphone = 0

    If Pinc.3 = 0 Then                          '=0, significa que esta pisado el freno
      Portb.4 = 1                              'Se Enciende ING
      Wait 2
      Portb.3 = 1                              'Se Enciende START
      Goto Starteng
    Else
      Input "B"                                  'Aviso para Smartphone, messagebox "falta aplastar
  Freno"
      Wait 2
    End If
  Goto Main

  Starteng:
    If Pinc.4 = 1 Then                          '= 1, significa que se levanta
      Portb.3 = 0                              'Apago START, Motor de Arranque
      Waitms 200
      Portb.5 = 1                              ' Se enciende Accesorios
    Else
      Goto Starteng
    End If
  Goto Main




  Startphone:
    Do
      Ischar = Inkey()
      Select Case Ischar
        Case &H72                               'busca identificar la letra "r" para parar START
```

```
      Cprevburn = 39000 + 1
    End Select

    Gosub Prevention                          'Contador para evitar que se queme Motor de
Arranque

  Loop Until Cprevburn > 39000

  Cprevburn = 0
  Portb.3 = 0                      'Apaga Motor de Arranque, identifico: letra "e" o levanto
bandera de prevencion
  Waitms 200
  Portb.5 = 1                      ' Se enciende Accesorios
Return

Prevention:
  Incr Cprevburn                   ' Disminuye el lapso para parar el Motor de Arranque
Return



Accessories:
  If Facesorio = 0 Then
    Portb.5 = 1
    Facesorio = 1
  Else
    Portb.5 = 0
    Facesorio = 0
  End If
Return



Keyfull:
  Portb.6 = 1
Return

Keyless:
  Portb.6 = 0
Return


Offengine:
  Portb.5 = 0
  Portb.4 = 0
  Portb.3 = 0
  Portb.7 = 0
  Waitms 500
Goto Main
```

Código implementado en la aplicación *smartphone*

```
 Public Class Form1

    Dim path As String
    Dim closetime As Byte
    Dim encrypass As String

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        path =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingA
ssembly().GetName().CodeBase)
        Try
            SerialPort1.Open()
        Catch
            MessageBox.Show("Mini Austin Out of Range", "Starting",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1)
        End Try
    End Sub


    Private Sub SendCharacter()
        Try
            SerialPort1.DiscardOutBuffer()
            SerialPort1.Write(Label2.Text)

        Catch ex As Exception
            SerialPort1.Close()

            Panel7.Visible = True   'Insert Password
            Panel7.BringToFront()
            TextBox6.Focus()

'Es para cuando se levante la ventana Insert Password, el  resto de
ventanas se
'deshabiliten y asi el cursor no corra a ventanas escondidas.
            Panel1.Enabled = False
            Panel2.Enabled = False
            Panel3.Enabled = False
            Panel4.Enabled = False
            Panel5.Enabled = False
            Panel6.Enabled = False
            Panel8.Enabled = False

            MenuItem2.Enabled = False   'Back
            MenuItem4.Enabled = False   'Password Setting
            MenuItem5.Enabled = False   'Car Keys Settings
        End Try
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
        If closetime = 0 Then
            Timer2.Enabled = False
```

```
'||Insert Password|| se esconde, y aparece ultima ventana activada.
            Panel7.SendToBack()
            Panel7.Visible = False

            With TextBox6
                .PasswordChar = ""
                .ForeColor = Color.DarkGray
                .Text = "Enter Password"
            End With
            Label7.Text = "..."

'Por motivos que se pidió la clave porque se desconecto el puerto
'entonces, se reactivan las ventanas una vez comprobada la clave.
            Panel1.Enabled = True
            Panel2.Enabled = True
            Panel3.Enabled = True
            Panel4.Enabled = True
            Panel5.Enabled = True
            Panel6.Enabled = True
            Panel8.Enabled = True

            MenuItem3.Enabled = False   'Start
            MenuItem2.Enabled = True    'Back
            MenuItem4.Enabled = True    'Password Setting
            MenuItem5.Enabled = True    'Car Keys Settings

        Else
            closetime = closetime - 1
        End If
    End Sub


    '|| Main Menu ||
    Private Sub Panel1_GotFocus(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Panel1.GotFocus

    End Sub
    '||Off Engine||
    Private Sub LinkLabel1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel1.Click
        Panel1.Visible = False       'Main Menu
        Panel2.Visible = True        'Off Engine
    End Sub
    '||On Engine||
    Private Sub LinkLabel2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel2.Click
        Dim ResultOnENG As DialogResult
        ResultOnENG = MessageBox.Show("Before START Engine, Check
GearBox to be on Neutral", "Before Start Engine",
MessageBoxButtons.OKCancel, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1)
        Select Case ResultOnENG
            Case Windows.Forms.DialogResult.Cancel
                Panel1.Visible = True
                Panel3.Visible = False
            Case Else
                Panel1.Visible = False
                Panel3.Visible = True
```

```
        End Select
    End Sub
    '||Accessories||
    Private Sub LinkLabel3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel3.Click
        Panel1.Visible = False
        Panel4.Visible = True
    End Sub
    '||Doors||
    Private Sub LinkLabel4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel4.Click
        Panel1.Visible = False
        Panel5.Visible = True
    End Sub


    '|| Off Engine ||
    Private Sub Panel2_GotFocus(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Panel2.GotFocus

    End Sub

    Private Sub CheckBox7_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles CheckBox7.KeyDown
        Label2.Text = "o"
        CheckBox7.Checked = True
        SendCharacter()
    End Sub

    Private Sub CheckBox7_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox7.Click
        CheckBox7.Checked = False
    End Sub

    Private Sub LinkLabel6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel6.Click
        BackPanel()
    End Sub


    '|| START Engine ||
    Private Sub Panel3_GotFocus(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Panel3.GotFocus

    End Sub

    Private Sub CheckBox5_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles CheckBox5.KeyDown
        CheckBox5.Checked = True
        Label2.Text = "e"
        SendCharacter()
    End Sub

    Private Sub CheckBox5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox5.Click
        Label2.Text = "r"
        CheckBox5.Checked = False
        SendCharacter()
```

```
    End Sub

    Private Sub LinkLabel7_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel7.Click
        BackPanel()
    End Sub


'|| Accessories ||
    Private Sub Panel4_GotFocus(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Panel4.GotFocus

    End Sub

    Private Sub CheckBox1_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles CheckBox1.KeyDown
        Label2.Text = "z"
        CheckBox1.Checked = True
        SendCharacter()
    End Sub

    Private Sub CheckBox1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox1.Click
        CheckBox1.Checked = False
    End Sub

    Private Sub LinkLabel8_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel8.Click
        BackPanel()
    End Sub


'|| Doors ||
    Private Sub Panel5_GotFocus(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Panel5.GotFocus

    End Sub

    Private Sub CheckBox3_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles CheckBox3.KeyDown
        Label2.Text = "a"
        CheckBox3.Checked = True
        SendCharacter()
    End Sub

    Private Sub CheckBox3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox3.Click
        CheckBox3.Checked = False
    End Sub

    Private Sub CheckBox4_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles CheckBox4.KeyDown
        Label2.Text = "c"
        CheckBox4.Checked = True
        SendCharacter()
    End Sub
```

```
    Private Sub CheckBox4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox4.Click
        CheckBox4.Checked = False
    End Sub

    Private Sub LinkLabel9_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel9.Click
        BackPanel()
    End Sub


'|| Password Setting ||
    Private Sub Panel6_GotFocus(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Panel6.GotFocus
    End Sub
'\\Comprobar adecuado ingreso, para cambiar Clave
    Private Sub TextBox1_KeyPress(ByVal sender As Object, _
    ByVal e As KeyPressEventArgs) Handles TextBox1.KeyPress
        Select Case e.KeyChar
            Case ChrW(13)
                ChangePassw()
                e.Handled = False
        End Select
    End Sub

    Private Sub LinkLabel10_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel10.Click
        ChangePassw()
    End Sub

    Private Sub ChangePassw()
        'Dim objreader As New System.IO.StreamReader(path + "\clave")
        Dim objreader As New System.IO.StreamReader(path +
"\Verschlusselung")

        encrypass = TextBox1.Text
        For Index As Byte = 1 To 5
            encrypass = MD5(encrypass)
        Next

        'Comparación, clave ingresada vs. clave grabada encriptada
        If objreader.ReadLine = encrypass Then
            Label6.Text = "Correct Password"

            With TextBox1
                .Visible = False
                .PasswordChar = ""
                .Text = "Type old Password"
            End With

            LinkLabel10.Visible = False

            TextBox2.Visible = True
            TextBox3.Visible = True
            LinkLabel11.Visible = True

            TextBox2.Focus()
        Else
```

```
            Label6.Text = "Incorrect Password"
            With TextBox1
                .Text = ""
                .Focus()
            End With
        End If
        objreader.Close()
    End Sub

'\\Etapa donde ingreso las 2 claves
    Private Sub TextBox3_KeyPress(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
        Select Case e.KeyChar
            Case ChrW(13)
                NewPassw()
                e.Handled = False
        End Select
    End Sub

    Private Sub LinkLabel11_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel11.Click
        NewPassw()
    End Sub

    Private Sub NewPassw()
        If (TextBox3.Text.Length Or TextBox4.Text.Length) >= 4 Then

            If TextBox2.Text = TextBox3.Text Then
                encrypass = TextBox2.Text

                For Index As Byte = 1 To 5
                    encrypass = MD5(encrypass)
                Next

                ']Dim objwriter As New System.IO.StreamWriter(path +
"\clave")
                ']objwriter.Write(TextBox2.Text)
                Dim objwriter As New System.IO.StreamWriter(path +
"\Verschlusselung")
                objwriter.Write(encrypass)
                objwriter.Close()

                Label6.Text = "New Password Saved"

                With TextBox2
                    .Visible = False
                    .Text = "Type Password"
                End With
                With TextBox3
                    .Visible = False
                    .Text = "Re-type Password"
                End With
                LinkLabel11.Visible = False

                With TextBox1
                    .Visible = True
                    .Focus()
                End With
```

```
                LinkLabel10.Visible = True

            Else
                Label6.Text = "Mismatched Password"
                With TextBox2
                    .Text = ""
                    .Focus()
                End With
                TextBox3.Text = ""
            End If

        Else
            Label6.Text = "Minimum 4 Characters"
            With TextBox2
                .Text = ""
                .Focus()
            End With
            TextBox3.Text = ""
        End If
    End Sub

'\\Decoración de Campos
    Private Sub TextBox1_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
        If TextBox1.Text = "Type old Password" Then
            TextBox1.Text = ""
        Else
            TextBox1.PasswordChar = "x"
            Label6.Text = "..."
        End If
    End Sub
    Private Sub TextBox1_LostFocus(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TextBox1.LostFocus
        If TextBox1.Text = "" Then
            TextBox1.PasswordChar = ""
            TextBox1.Text = "Type old Password"
            Label6.Text = "..."
        End If
    End Sub
'\\Decoración de Campos
    Private Sub TextBox2_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles TextBox2.KeyDown
        If TextBox2.Text = "Type Password" Then
            TextBox2.Text = ""
        Else
            Label6.Text = "..."
        End If
    End Sub
    Private Sub TextBox2_LostFocus(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TextBox2.LostFocus
        If TextBox2.Text = "" Then
            TextBox2.Text = "Type Password"
        End If
    End Sub
'\\Decoración de Campos
    Private Sub TextBox3_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles TextBox3.KeyDown
        If TextBox3.Text = "Re-type Password" Then
```

```
            TextBox3.Text = ""
        Else
            Label6.Text = "..."
        End If
    End Sub
    Private Sub TextBox3_LostFocus(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TextBox3.LostFocus
        If TextBox3.Text = "" Then
            TextBox3.Text = "Re-type Password"
        End If
    End Sub
'\\Linklabel_Back de Password Setting
    Private Sub LinkLabel14_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel14.Click
        Panel6.Visible = False
        Label6.Text = "..."

        TextBox2.Visible = False
        TextBox3.Visible = False
        LinkLabel11.Visible = False

        With TextBox2
            .Text = "Type Password"
        End With
        With TextBox3
            .Text = "Re-type Password"
        End With

        With TextBox1
            .Visible = True
        End With
        LinkLabel10.Visible = True

        Panel1.Visible = True       'Main Menu
        MenuItem2.Enabled = True    'Back_Menu Item
    End Sub


'|| Para Grabar por 1ra vez la clave
    Private Sub LinkLabel12_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel12.Click
        encrypass = TextBox4.Text
        For Index As Byte = 1 To 5
            encrypass = MD5(encrypass)
        Next
        Dim objwriter As New System.IO.StreamWriter(path +
"\Verschlusselung")
        objwriter.Write(encrypass)
        objwriter.Close()
    End Sub

'|| Función Publica ||
    Public Function MD5(ByRef strText As String) As String
        Dim MD5Service As New
System.Security.Cryptography.MD5CryptoServiceProvider
        Dim Bytes() As Byte =
MD5Service.ComputeHash(System.Text.Encoding.ASCII.GetBytes(strText))
        Dim S As String
```

```
        For Each By As Byte In Bytes
            S += By.ToString("x2")
        Next
        Return S
    End Function


'|| Insert Password ||
    Private Sub Panel7_GotFocus(ByVal sender As System.Object, ByVal e
As System.EventArgs)

    End Sub

    Private Sub TextBox6_KeyPress(ByVal sender As Object, _
    ByVal e As KeyPressEventArgs) Handles TextBox6.KeyPress
        Select Case e.KeyChar
            Case ChrW(13)
                EnterPassw()
                e.Handled = False
        End Select
    End Sub

    Private Sub LinkLabel13_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel13.Click
        EnterPassw()
    End Sub

    Private Sub EnterPassw()
        ']Dim objreader As New System.IO.StreamReader(path + "\clave")
        Dim objreader As New System.IO.StreamReader(path +
"\Verschlusselung")

        encrypass = TextBox6.Text
        For Index As Byte = 1 To 5
            encrypass = MD5(encrypass)
        Next

        If objreader.ReadLine = encrypass Then
            ']If objreader.ReadLine = TextBox6.Text Then
            Label7.Text = "Welcome"
            closetime = 1
            Timer2.Enabled = True

            If SerialPort1.IsOpen() Then
'Do Nothing (pienso que es True y existe conexión)
            Else
'Se va por este camino, porque no hay conexión
                Try
                    SerialPort1.Open()
                Catch
                    MessageBox.Show("Mini Austin Out of Range",
"Connection Status", MessageBoxButtons.OK, MessageBoxIcon.Asterisk,
MessageBoxDefaultButton.Button1)
                End Try
            End If

        Else
```

```
            Label7.Text = "Incorrect Password"
            With TextBox6
                .Text = ""
                .Focus()
            End With
        End If
        objreader.Close()
    End Sub
'\\Decoración de Campos
    Private Sub TextBox6_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles TextBox6.KeyDown
        If TextBox6.Text = "Enter Password" Then
            With TextBox6
                .Text = ""
                .PasswordChar = "x"
            End With
        Else
            TextBox6.ForeColor = Color.Black
            Label7.Text = "..."
        End If
    End Sub
    Private Sub TextBox6_LostFocus(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TextBox6.LostFocus
        If TextBox6.Text = "" Then
            With TextBox6
                .PasswordChar = ""
                .ForeColor = Color.DarkGray
                .Text = "Enter Password"
            End With
            Label7.Text = "..."
        End If
    End Sub


'|| Manual Start ||
    Private Sub Panel8_GotFocus_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Panel8.GotFocus

    End Sub

    Private Sub CheckBox2_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles CheckBox2.KeyDown
        Label2.Text = "k"
        CheckBox2.Checked = True
        SendCharacter()
    End Sub
    Private Sub CheckBox2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox2.Click
        CheckBox2.Checked = False
    End Sub


    Private Sub CheckBox6_KeyDown(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyEventArgs) Handles CheckBox6.KeyDown
        Label2.Text = "l"
        CheckBox6.Checked = True
        SendCharacter()
    End Sub
```

```
    Private Sub CheckBox6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox6.Click
        CheckBox6.Checked = False
    End Sub

'\\Back
    Private Sub LinkLabel15_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel15.Click
        BackPanel()
    End Sub


'~~ Serial Port ~~
    Private Sub SerialPort1_DataReceived(ByVal sender As System.Object,
ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
        ReceivedText(SerialPort1.ReadExisting())
    End Sub

    Private Sub ReceivedText(ByVal [text] As String)
      'Compares the ID of the creating Thread to the ID of the calling
      Thread
        If TextBox7.InvokeRequired Then
            Dim x As New SetTextCallback(AddressOf ReceivedText)
            Me.Invoke(x, New Object() {(text)})
        Else
            TextBox7.Text &= [text]
            Select Case text
                Case "m"
                    SerialPort1.Write("n")
                    SerialPort1.DiscardOutBuffer()

                Case "b"
                    MessageBox.Show("Push Brake", "Mini Austin",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk,
MessageBoxDefaultButton.Button1)
            End Select
            text = ""
            SerialPort1.DiscardInBuffer()
        End If
    End Sub
    Delegate Sub SetTextCallback(ByVal [text] As String)
    'Added to prevent threading errors during receiveing of data

'||Sección de Menu_Items||
'|| Back ||
    Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem2.Click
        BackPanel()
    End Sub

    Private Sub BackPanel()
        Panel2.Visible = False
        Panel3.Visible = False
        Panel4.Visible = False
        Panel5.Visible = False
        Panel6.Visible = False
        Panel7.Visible = False
```

```
        Panel8.Visible = False

        Panel1.Visible = True   'Menu Principal
    End Sub

'|| START (Insert Password) ||
    Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem3.Click
        With Panel1
            .Visible = True
            .Enabled = False
        End With

        With Panel7
            .Visible = True
            .BringToFront()
            .Focus()
        End With
    End Sub

'|| Password Setting ||
    Private Sub MenuItem4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem4.Click
        Panel1.Visible = False
        Panel2.Visible = False
        Panel3.Visible = False
        Panel4.Visible = False
        Panel5.Visible = False
        Panel7.Visible = False
        Panel8.Visible = False

        With Panel6
            .Visible = True
            .Focus()
        End With
        MenuItem2.Enabled = False   'Back
    End Sub

'|| Car Keys Settings ||
    Private Sub MenuItem5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem5.Click
        Panel1.Visible = False
        Panel2.Visible = False
        Panel3.Visible = False
        Panel4.Visible = False
        Panel5.Visible = False
        Panel6.Visible = False
        Panel7.Visible = False

        With Panel8
            .Visible = True
            .Focus()
        End With
    End Sub

'|| Exit ||
    Private Sub MenuItem6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem6.Click
```

```
        Me.Close()
    End Sub
End Class
```